

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2003-022621

(43)Date of publication of application : 24.01.2003

(51)Int.Cl. G11B 20/12
G11B 20/10
G11B 27/00
H04N 5/92

(21)Application number : 2001-207244 (71)Applicant : SHARP CORP

(22)Date of filing : 09.07.2001 (72)Inventor : KIYAMA JIRO
IWANO HIROTOSHI
YAMAGUCHI TAKAYOSHI

(54) METHOD FOR RECORDING DATA, METHOD FOR MODIFYING DATA AND
APPARATUS THEREFOR

(57)Abstract:

PROBLEM TO BE SOLVED: To facilitate the updating of management information multiplexed in an AV stream.

SOLUTION: A method for recording data is 4 for recording, on a recording medium, a first unit (Record Unit) to store first data consisting of a video or a sound and second data to be reproduced by synchronizing with the first data, and management information (Movie fragment atom) to manage the first unit. The first unit and the management information are disposed closely to each other on the recording medium.

* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.**** shows the word which can not be translated.

3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1]The 1st unit that stores the 1st data that consists of an image or a sound, and the 2nd data reproduced synchronizing with said 1st data, A data recording method which is a data recording method which records management information which manages said 1st unit on a recording medium, and is characterized by arranging said 1st unit and said management information to the neighborhood on said recording medium.

[Claim 2]The 1st unit that stores the 1st data that consists of an image or a sound, and the 2nd data reproduced synchronizing with said 1st data, The 1st management information that manages said 1st data, and the 2nd management information that manages said 2nd data, A data recording method which is a data recording method recorded on a recording medium, and is characterized by arranging mutually said 2nd one or more management information to the neighborhood on said recording medium while separating and arranging said 2nd management information and said 1st management information on said recording medium.

[Claim 3]The 1st unit that stores the 1st data that consists of an image or a sound, and the 2nd data reproduced synchronizing with said 1st data, A data changing method which is the data changing method that management information which manages said 1st unit rewrites said 2nd data and said management information to a recording medium which has been arranged and was recorded on the neighborhood, and is characterized by rewriting said 2nd data and said management information continuously.

[Claim 4]The 1st unit that stores the 1st data that consists of an image or a sound, and the 2nd data reproduced synchronizing with said 1st data, It is recorded by the 1st management information that manages said 1st data, and further, A data changing method which is the data changing method which rewrites said 2nd data and said 2nd

management information, and is characterized by rewriting said 2nd management information continuously to a recording medium with which one or more of said 2nd management information were mutually arranged and recorded on the neighborhood.

[Claim 5]The 1st unit that stores the 1st data that consists of an image or a sound, and the 2nd data reproduced synchronizing with said 1st data, A data recorder which is a data recorder which records management information which manages said 1st unit on a recording medium, and is characterized by having a means to arrange and record said management information near said 1st unit on said recording medium.

[Claim 6]The 1st unit that stores the 1st data that consists of an image or a sound, and the 2nd data reproduced synchronizing with said 1st data, The 1st management information that manages said 1st data, and the 2nd management information that manages said 2nd data, A data recorder which is a data recorder recorded on a recording medium, and is characterized by having a means to arrange and record mutually said 2nd one or more management information on the neighborhood on said recording medium while having separated and arranged said 2nd management information and said 1st management information on said recording medium.

[Claim 7]The 1st unit that stores the 1st data that consists of an image or a sound, and the 2nd data reproduced synchronizing with said 1st data, A data changing device which is a data changing device with which management information which manages said 1st unit rewrites said 2nd data and said management information to a recording medium which has been arranged and was recorded on the neighborhood, and is characterized by having a means which rewrites said management information continuously with said 2nd data.

[Claim 8]The 1st unit that stores the 1st data that consists of an image or a sound, and the 2nd data reproduced synchronizing with said 1st data, It is recorded by the 1st management information that manages said 1st data, and further, A data changing device which is a data changing device which rewrites said 2nd data and said 2nd management information, and is characterized by having a means which rewrites said 2nd management information continuously to a recording medium with which one or more of said 2nd management information were mutually arranged and recorded on the neighborhood.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Field of the Invention]This invention relates to the data recording method and the data changing method of recording and changing picture image data and voice data,

and its equipment to the recording medium in which random access, such as a hard disk and an optical disc, is possible.

[0002]

[Description of the Prior Art]The digital recording playback equipment of the video using a disk medium is spreading. In a disk medium, unlike a tape medium, access to arbitrary parts is possible in a short time, and uses the character, The function called the non-destroying edit or non-linear editing which edits only by rewriting of management information on the disk of one sheet without copying data, and the function which records the video over the dispersed free space after deleting without overwriting recorded data are realizable.

[0003]In the video recording equipment using a disk medium, it is common to summarize management information at the end and to record it at the time of recording. However, when a power supply is shut off suddenly and management information is not able to be recorded during recording, it becomes impossible to manage the data recorded by then, and there is a problem of being playback impossible.

[0004]In order to cope with this problem, the concept of Fragmented movie is introduced, for example by Motion JPEG 2000. Here, the outline is explained about Fragmented movie. The typical composition of Motion JPEG 2000 file containing Fragmented movie is shown in drawing 32.

[0005]Movie data atom which Movie atom which manages the information common to the whole file at the head is arranged, and stores partial AV stream data (referred to as Movie fragment) after that, Movie fragment atom which manages the Movie fragment is arranged by turns.

[0006]By recording in this turn at the time of recording, Movie fragment atom recorded immediately before even when a power supply was temporarily shut off during recording, and Movie fragment which it manages remain in the disk, and it becomes possible to play later.

[0007]

[Problem to be solved by the invention]Also in a disk medium, a postrecording function is required like a tape medium. As an AV stream corresponding to a postrecording function, as shown in drawing 33, what multiplexed the field for storing data at the time of postrecording can be considered.

[0008]Also about the postrecorded data, it is a user's property and to protect as much as possible by backup of management information is desired. However, in Motion JPEG2000 above-mentioned standard, there is no composition of the management information about such a stream.

[0009]This invention], In light of the above-mentioned problems, this invention is a thing.

The purpose is to provide the data recording method which enables protection of the

management information about FUREKO data, and DE.

[0010]

[Means for solving problem]The 1st unit in which the 1st invention of an application concerned stores the 1st data that consists of an image or a sound, and the 2nd data reproduced synchronizing with said 1st data, It is a data recording method which records the management information which manages said 1st unit on a recording medium, and said 1st unit and said management information are arranged to the neighborhood on said recording medium.

[0011]The 1st unit in which the 2nd invention of an application concerned stores the 1st data that consists of an image or a sound, and the 2nd data reproduced synchronizing with said 1st data, The 1st management information that manages said 1st data, and the 2nd management information that manages said 2nd data, While being a data recording method recorded on a recording medium and separating and arranging said 2nd management information and said 1st management information on said recording medium, said 2nd one or more management information is mutually arranged to the neighborhood on said recording medium.

[0012]The 1st unit in which the 3rd invention of an application concerned stores the 1st data that consists of an image or a sound, and the 2nd data reproduced synchronizing with said 1st data, To the recording medium which has been arranged and was recorded on the neighborhood, the management information which manages said 1st unit is the data changing method which rewrites said 2nd data and said management information, and rewrites said 2nd data and said management information continuously.

[0013]The 1st unit in which the 4th invention of an application concerned stores the 1st data that consists of an image or a sound, and the 2nd data reproduced synchronizing with said 1st data, It is recorded by the 1st management information that manages said 1st data, and further, To the recording medium which has been arranged and was mutually recorded on the neighborhood one or more pieces, said 2nd management information is the data changing method which rewrites said 2nd data and said 2nd management information, and rewrites said 2nd management information continuously.

[0014]The 1st unit in which the 5th invention of an application concerned stores the 1st data that consists of an image or a sound, and the 2nd data reproduced synchronizing with said 1st data, It is a data recorder which records the management information which manages said 1st unit on a recording medium, and had a means to arrange and record said management information near said 1st unit on said recording medium.

[0015]The 1st unit in which the 6th invention of an application concerned stores the 1st data that consists of an image or a sound, and the 2nd data reproduced

synchronizing with said 1st data, The 1st management information that manages said 1st data, and the 2nd management information that manages said 2nd data, While being a data recorder recorded on a recording medium and having separated and arranged said 2nd management information and said 1st management information on said recording medium, it had a means to arrange and record mutually said 2nd one or more management information on the neighborhood on said recording medium.

[0016]The 1st unit in which the 7th invention of an application concerned stores the 1st data that consists of an image or a sound, and the 2nd data reproduced synchronizing with said 1st data, It is a data changing device with which the management information which manages said 1st unit rewrites said 2nd data and said management information to the recording medium which has been arranged and was recorded on the neighborhood, and had the means which rewrites said management information continuously with said 2nd data.

[0017]The 1st unit in which the 8th invention of an application concerned stores the 1st data that consists of an image or a sound, and the 2nd data reproduced synchronizing with said 1st data, It is recorded by the 1st management information that manages said 1st data, and further, To the recording medium with which one or more of said 2nd management information were mutually arranged and recorded on the neighborhood, it is a data changing device which rewrites said 2nd data and said 2nd management information, and had the means which rewrites said 2nd management information continuously.

[0018]

[Mode for carrying out the invention]Hereafter, the embodiment of this invention is described in detail, referring to Drawings.

[0019]<System configuration> drawing 1 is a block diagram of the video disc recorder which can be postrecorded used in common in this embodiment. As shown in drawing 1, this equipment, The bus 100, the host CPU 101, RAM102, ROM103, the user interface 104, the system clock 105, the optical disc 106, the pickup 107, ECC decoders 108, ECC encoder 109, the buffer 110 for playback, The buffer 111 for record/postrecording, the demultiplexer 112, the multiplexer 113, the buffer 114 for multiplexing, the audio decoder 115, the video decoder 116, the audio encoder 117, the video encoder 118 and the camera that is not illustrated, a microphone, It comprises a loudspeaker, a display, etc.

[0020]the host CPU 101 leads the bus 100 -- the demultiplexer 112, the multiplexer 113, and the pickup 107 -- although not illustrated, communication with the audio decoder 115, the video decoder 116, the audio encoder 117, and the video encoder 118 is performed.

[0021]At the time of playback, the error correction of the data read from the optical disc 106 through the pickup 107 is carried out by ECC decoders 108, and it is once stored in the buffer 110 for playback. The demultiplexer 112 distributes the data in the

buffer for reproduction to a suitable decoder by the classification according to the data transmission request from the audio decoder 115 and the video decoder 116.

[0022]On the other hand, at the time of record, the data by which compression encoding was carried out with the audio encoder 117 and the video encoder 118 is once sent to the buffer 114 for multiplexing, and AV multiplexing is carried out by the multiplexer 113 and it is sent to the buffer 111 for record/postrecording. By ECC encoder 109, an error correcting code is added to the data in the buffer 111 for record/postrecording, and it is recorded on the optical disc 106 through the pickup 107.

[0023]MPEG-1 Layer-II is used for the coding mode of audio information, and MPEG-2 is used for the coding mode of a video data, respectively.

[0024]Let the optical disc 106 be an optical disc in which record reproduction is spirally performed toward inner circumference from a periphery and which can be desorbed. 2048 bytes is used as one sector and an ECC block consists of 16 sectors for an error correction. When rewriting the data in an ECC block, it is necessary to read the whole ECC block in which the data is contained, to perform an error correction, and to rewrite the target data, and it necessary to add an error correcting code again, to constitute an ECC block, and to record on a recording medium. In order that the optical disc 106 may raise recording efficiency, ZCAV (zone constant angular velocity) is adopted, and a record section comprises several zones where number of rotations differs.

[0025]A file system is used in order to manage the variety of information on the <file system> optical disc 106. In consideration of interoperability with a personal computer (PC), UDF (Universal Disk Format) is used for a file system. On a file system, various management information and an AV stream are treated as a file.

[0026]User area is managed by 2048 bytes of logical block (a sector and a one to one correspondence). Each file comprises an extent (continuous logical block) of integer pieces, and it may distribute per extent and it may be recorded. Free space is managed per logical block using Space Bitmap.

[0027]A QuickTime file format is used as a format for <file format> AV stream management. A QuickTime file format is a format for multimedia data management which Apple developed, and it is used widely in the world of PC.

[0028]A QuickTime file format comprises a video data, audio information (these are named generically and it is also called media data), etc. and management information. Both are doubled and it is called a QuickTime movie (carrying out abbreviated movie) here. Both may exist in the same file or may exist in a separate file.

[0029]When it exists in the same file, composition as shown in drawing 2 (a) is taken. A variety of information is stored in a common structure of atom. Management information is stored in the structure of Movie atom, and an AV stream is stored in the structure of Movie data atom. The table for drawing the relative position of a under

[the file of the AV information corresponding to the arbitrary time in media data], the attribution information of media data, the external reference information mentioned later, etc. are included in the management information in Movieatom.

[0030]On the other hand, when management information and media data are stored in a separate file, composition as shown in drawing 2 (b) is taken. An AV stream does not need to be stored in atom although management information is stored in the structure of Movie atom. At this time, it is said that Movie atom is carrying out "external reference" of the file which stored the AV stream.

[0031]External reference can be carried out to two or more AV stream files, as shown in drawing 2 (c), and what is called "non-linear editing" and "non-destroying edit" which are shown as edited seemingly are attained according to this structure, without moving the AV stream itself physically.

[0032]Then, the format of the management information of QuickTime is explained using drawing 3 thru/or drawing 12. First, atom which is a common information storing format is explained. At the head of atom, Atom size which is the size of the atom, and Type which is the type information of the atom certainly exist. Type is distinguished by four characters, for example, it is set to 'moov' in Movie atom, and it serves as 'mdat' in Movie data atom.

[0033]Each atom can contain another atom. That is, a layered structure is between atom(s). The composition of Movie atom is shown in drawing 3. Movie header atom manages the overall attribute of the movie which the Movie atom manages. Track atom stores the information about tracks included in the movie, such as video and an audio. User data atom is atom which can be defined uniquely.

[0034]The composition of Track atom is shown in drawing 4. Track header atom manages the overall attribute of the track. Edit atom manages which section of media data is reproduced in which timing of a movie. Track reference atom manages a relation with another track. Media atom manages data called actual video and audio.

[0035]The composition of Track header atom is shown in drawing 5. Here, only a thing required for next explanation is explained. flags is a set of the flag which shows an attribute. There is a Track enabled flag, as a typical thing, if this flag is 1, that track will be reproduced, and it will not be reproduced if it is 0. layer expresses the spatial priority of the track and a picture is displayed for the track where the value of layer with plurality is [the track which displays a picture] smaller on a front face.

[0036]The composition of Media atom is shown in drawing 6. Media header atom manages the overall attribute about the media data which the Media atom manages, etc. Handler referenceatom stores the information which shows by which decoder media data is decoded. Media information atom manages attribution information peculiar to media, such as video and an audio.

[0037]The composition of Media information atom is shown in drawing 7. Media information header atom manages attribution information peculiar to media, such as

video and an audio. Handler reference atom is as the clause of Media atom having explained. Data information atom contains Data reference atom which is atom which manages the name of the file containing the media data which the QuickTime movie refers to. Sample table atom has managed size, reproducing time, etc. of data.

[0038]Next, although Sample table atom is explained, the management method of the data in QuickTime is explained before that using drawing 8. In QuickTime, the minimum unit (for example, video frame) of data is called a sample. The sample is numbered from 1 in order of reproducing time for each track of every (sample number).

[0039]In the QuickTime format, each reproducing time length and data size of a sample are managed. The sample belonging to the same track calls a chunk the field continuously arranged in a file at the order of reproducing time. The chunk is also numbered from 1 in order of reproducing time.

[0040]In the QuickTime format, the measurement size which each address and each chunk from the file head of a chunk contain is managed. It is possible to search for the position of the sample corresponding to arbitrary time based on these information.

[0041]The composition of Sample table atom is shown in drawing 9. Sample description atom manages the index of the chunk of a file etc. with which each data format (Data format) and sample of the chunk are stored. Time-to-sample atom manages the reproducing time of each sample.

[0042]Sync sample atom manages the sample in which a decoding start is possible among each samples. Sample-to-chunk atom manages the measurement size contained in each chunk. Sample size atom manages the size of each sample. Chunk offset atom manages the address from the file head of each chunk.

[0043]Edit atom contains one Edit list atom, as shown in drawing 10. Edit list atom has a group (entry) of the value of Track duration for the number specified by Number of entries, Media time, and Media rate. On the track, each entry corresponded to the section reproduced continuously, and is located in a line in order of the reproducing time on the track.

[0044]The reproducing time on the track of the section when the entry manages Track duration, the position on the media data corresponding to the head of the section when the entry manages Media time, and Media rate express the reproduction speed of the section which the entry manages. When Media time is -1, reproduction of the sample in a part for Track duration of the entry and its track is suspended. This section is called empty edit.

[0045]The example of use of Edit list is shown in drawing 11. Here, the contents of Edit list atom are contents shown in drawing 11 (a), and suppose that the composition of the sample was drawing 11 (b) further. Here, $D(i)$ and Media time are made into $T(i)$ and Media rate is made into $R(i)$ for Track duration of the i -th entry. At this time, reproduction of a actual sample is performed in the order shown in drawing 11 (c). This is explained briefly.

[0046]First, since 13000 and Media time are [20000 and Media rate] 1 entry #1 in Track duration, the section of 13000 reproduces the section of the time 20000–33000 in a sample from the head of the track. Next, since Track duration is [5000 and Mediatime] –1, as for the section of the time 13000–18000 in a track, and nothing, entry #2 is reproduced.

[0047]Finally, since 10000 and Media time are [0 and Media rate] 1, in the section of the time 18000–28000 in a track, as for entry #3, Track duration reproduces the section of the time 0–10000 in a sample.

[0048]The composition of User data atom is shown in drawing 12. Arbitrary number storing of the original information which is not defined by QuickTime format can be carried out at this atom. One original information is managed by one entry, and one entry comprises Size, Type, and User data. Size expresses the size of the entry itself and identification information for Type to distinguish original information, respectively and User data express actual data.

[0049]Next, in order to correspond to the power supply cutoff under recording, etc., Fragmented Movie which is the concept introduced into the QuickTime file format is explained. Fragmented movie is the concept introduced by Motion JPEG2000 which is 1 application of a QuickTime format, It is possible to manage the information equivalent to above-mentioned Sample table atom for every partial AV stream.

[0050]The entire configuration of the QuickTime file which introduced Fragmented movie is shown in drawing 13. Movie data atom which Movie atom which manages the information common to the whole file is arranged, and stores Movie fragment in a head after that, Movie fragment atom which manages the address of a sample, size, reproducing time which constitute the Movie fragment is arranged by turns. AV stream data may exist in another file like the usual QuickTime file.

[0051]By recording in this turn at the time of recording, it is possible to prevent the damage caused by powering off at the time of recording to the minimum. Movie extends atom for that the QuickTime movie is Fragmented movie to be shown is contained in Movie atom. The default value about each track included in the movie is stored in Movie extends atom.

[0052]The management information about Moviefragment which the Movie fragment atom manages is contained in Movie fragment atom. There are Movie fragment header atom which stores the information about the whole Moviefragment to manage, and Track fragment atom which stores the information about each track in Movie fragment in management information.

[0053][0053]. Track fragmen is provided with the following.

Track fragment heade which stores the information about Movie fragment belonging to the track with which it manages atom.

Track fragment ru which manages the logical continuation field (referred to as Track run) which constitutes atom and Movie fragment belonging to the track, respectively.

Below, each atom is explained in detail.

[0054]The composition of Movie extends atom is shown in drawing 14. Movie extends atom has a role which shows as mentioned above that the QuickTime movie containing this atom is Fragmented movie.

[0055]The composition of Track extends atom is shown in drawing 15. Track extends atom exists in order to set up the default value of the sample of each track included in this QuickTime movie. Refer to track ID of the track defined in Movie atom for track ID. The field which starts with default-sample- sets up the default value of track fragment managed by this atom.

[0056]The composition of Movie fragment atom is shown in drawing 16. The management information recorded one by one during recording is this atom. This atom contains Movie fragment header atom and Track fragment atom which are atom which stores the actual information about Movie fragment which this atom manages as above-mentioned.

[0057]The composition of Movie fragment header atom is shown in drawing 17. The main information stored in this atom is sequence-number. sequence-number expresses the turn from the head of Movie fragment which Movie fragment atom in which this atom is contained manages.

[0058]The composition of Track fragment atom is shown in drawing 18. Track fragment atom stores Track fragment header atom and Track fragment run atom which are the management information about the sample of the specific track included in Moviefragment.

[0059]The composition of Track fragment header atom is shown in drawing 19. This atom stores the default value about the sample of the specific track included in Movie fragment, etc. track-ID shows correspondence with track ID of the track defined in Movie atom. sample-description-index, The index number of sampledescription table which the sample which this atom manages refers to, and the field which starts with default-sample are the default values of the sample which this atom manages, respectively.

[0060]The composition of Track fragment run atom is shown in drawing 20. This atom stores the management information of the continuation field which is called Track run and which this atom manages, or each sample. sample-count shows the number of the sample contained in Track run. data-offset shows the offset value of Track run from base-data-offset. The field which starts with sample- stores values, such as reproducing time etc. of the sample which this atom manages. However, if the same as an above-mentioned default value, it is possible to omit and to reduce data size.

[0061]In order to manage the QuickTime movie contained in a <index file> disk, a special QuickTime movie file called AV index file is placed into [one] a disk.

[0062]The thumbnail and the various attributes about files (Still Picture Sub-Division currently referred to from the QuickTime movie or the QuickTime movie) in a disk are

registered into AV index file. link count which shows the number of times to which external reference of the file is carried out is in various attributes.

[0063]It can prevent deleting carelessly the file which can know easily whether there is any file which is referring to the file, and is referred to from others by referring to link count.

[0064]One working example of <working-example> this invention is described using drawing 21 thru/or drawing 31.

[0065]<the form of an AV stream> -- the composition of the AV stream in this example is first explained using drawing 21 and drawing 22. An AV stream comprises Record Unit (RU) of integer pieces. RU is a unit continuously recorded on a disk. The length of RU, RU which constitutes an AV stream is arranged on a disk how. It is set up so that ** seamless playback (it can play without a picture and a sound breaking off during playback), and real-time postrecording (record an audio, carrying out seamless playback of the video for postrecording) may be guaranteed. This setting method is mentioned later.

[0066]A stream is constituted so that RU boundary may be in agreement with an ECC block boundary. Arrangement of RU unit can be easily changed on a disk, guaranteeing seamless playback with these character of RU, even after recording an AV stream on a disk.

[0067]RU comprises Video Unit (VU) and Post Recording Unit (PRU) of integer pieces. VU is an independent refreshable unit and can serve as an entry point in the case of reproduction from that. PRU is a field for recording the data which carries out synchronous reproduction to VU contained in the same RU.

[0068]VU composition is shown in drawing 22. VU comprises GOP (group OBU picture) of the integer pieces which stored the video data for about 1 second, and AAU (audio access unit) of the integer pieces which stored the main audio information reproduced at the same time as them.

[0069]GOP is a unit of the graphical data compression in MPEG-2 video standard, and comprises two or more video frames (typically about 15 frames). AAU is a unit of the speech compression in an MPEG-1 Layer-II standard, and is constituted by the sound-wave-forms sample point of 1152 points. When a sampling frequency is 48 kHz, the reproducing time per AAU will be 0.024 second. In VU, in order to make small delay which is needed for AV synchronous reproduction, it arranges in order of AAUGOP.

[0070]In order to enable independent reproduction per VU, Sequence Header (SH) is placed at the head of the video data in VU, and Sequence End Code (SEC) is placed at an end. The reproducing time of VU is defined as having applied the video frame cycle to the number of video frames contained in VU. The end of VU is filled up with 0 in order to double the always end of RU with an ECC block boundary, when it constitutes RU combining several ready VU.

[0071]The area size of PRU should apply the maximum bit rate of the audio to the

reproducing time of RU here.

[0072]<AV stream management method> The composition of the management information of the above-mentioned AV stream is explained. An AV stream is managed with the movie file 2401 and the management information backup file 2402 equivalent to backup of the management information of the movie file 2401, as shown in drawing 23.

[0073]The movie file 2401 comprises an AV stream stored in Movie data atom, and Movie atom which manages the address of a sample, size, reproducing time which constitute an AV stream. An AV stream comprises an RU as mentioned above, and each RU is recorded as certainly being continuously arranged on a disk.

[0074]On the other hand, the management information backup file 2402 comprises Movie fragment atom which manages each RU in the movie file 2401. On a disk, these two files are multiplexed per RU, as shown in drawing 23. The Reason for taking such composition is explained below.

[0075]When composition like drawing 13 is taken, although it is convenient in an optical disc at the time of recording, at the time of playback, it is inconvenient. Because, the management information of the time to being recordable on Movie fragment atom, without seeking at the time of recording at the time of playback. It is because it is necessary to read Movie fragment atom#1 first so in order to reproduce Movie data atom#1, and it must be made to go back and forth between a pickup and the efficiency of reading worsens.

[0076]By taking composition as shown in drawing 23 at this time, while the backup at the time of breakage of the power off under recording or Movie atom of the movie file 2401 is attained, Reading of the efficient management information at the time of reproduction becomes possible like the case where Movie fragment atom is not used.

[0077]in addition -- although the two above-mentioned files are multiplexed and recorded on a disk, it is not necessary to record the whole continuously on a disk -- every -- Record Unit and last Movie fragment should just be recorded continuously. That is, except the discontinuity possible point shown in drawing 24, it records continuously on a disk. The Reason for performing such continuous recording is because renewal of Movie fragment which is needed with postrecording at the time of postrecording just before recording on PRU is enabled without adding seeking.

[0078]Although drawing 23 is indicated that Movie fragment atom and RU have touched, Actually, between Movie fragment atom and RU, Free space atom is inserted as padding so that the number of bytes from the head of Movie fragment atom to just before RU may become an integral multiple of 1ECC block size. Free spaceatom is atom specified in the QuickTime file format, and it is used for reservation of a field.

[0079]Next, the composition of the management information of the movie file 2401 is explained. The management information of the movie file 2401 comprises a postrecording audio track for managing the video track for video datas, the main audio

track for main audios, the postrecording field track for postrecording field management, and the actually postrecorded data.

[0080]A video track manages the lump of the video in one sample and VU for the video data from Sequence header to Sequence end code as one chunk. A main audio track manages the lump of the audio in one sample and VU for AAU as one chunk. A postrecording field track manages AAU as what recorded audio information on PRU by the predetermined bit rate, and manages one sample and PRU as one chunk.

[0081]However, it is for telling where of which timing this track records audio information, and flag of Track header atom is set up improper [reproduction] not reproduce accidentally. A postrecording audio track manages one sample and the continuous recording data in PRU for 1AAU as one chunk about the audio information actually recorded on PRU.

[0082]Next, the composition of the management information of the management information backup file 2402 is explained. Like the movie file 2401, a video track, a main audio track, It constitutes from a postrecording field track and a postrecording audio track, and one sample and the video data from Sequence header to Sequence end code are managed for AAU as one sample.

[0083]The management method of PRU in a backup management information file is explained taking the case of RU of composition of being shown in drawing 25. Here, RU presupposes that it is in L byte's position from the head of a movie file. PRU comprises ten AAU(s) and four AAU(s) make finishing [postrecording] un-postrecording and six AAU(s) of the second half from a head. And suppose again that each AAU is reproducing time 0.024 second, and size is 768 bytes.

[0084]At this time, as shown in drawing 26, default-sample-size of the postrecording field track in Movie extends atom and a postrecording audio track is 768, and default-sample-duration is 2160 (Timescale which is the fineness for 1 second.). It becomes a case where it is referred to as 90000.

[0085]The contents of the postrecording field track in Movie fragment atom which manages this Movie fragment, and the postrecording audio track become as they are shown in drawing 27. First, sample-count is set to ten in order to manage all the samples in PRU about a postrecording field track.

[0086]Next, in order to manage only a postrecorded sample about a postrecording audio track, sample-count is set to six and data-offset serves as a number of bytes from the head of PRU to a postrecorded head sample. When postrecording, these values of a postrecording audio track will be rewritten.

[0087]<Record Unit reproducing time deciding method> The deciding method of RU reproducing time in the stream corresponding to postrecording is explained first. The postrecording algorithm (reference postrecording algorithm) which serves as a device (reference device model) used as a standard and a standard in this deciding method for the compatibility reservation between apparatus is assumed, Next, RU reproducing

time is decided so that seamless reproduction may not fail, when it postrecords using them.

[0088]Then, a reference device model is first explained using drawing 28. A reference device model, One pickup, the ECC encoder decoder 501 connected with it, the track buffer 502, the demultiplexer 503, the buffer 504 for postrecording, the audio encoder 509, the video buffer 505, the audio buffer 506, the video decoder 507, It comprises the audio decoder 508.

[0089]In this model, since a pickup is one piece, time sharing performs recording on the disk of read-out from the disk of the data for playback, and the data for postrecording. When reading the data for playback from a disk, it reads including PRU and last Movie fragment atom. Read Movie fragment atom and the ECC block (PRU block) containing PRU are sent to the buffer 504 for postrecording from the track buffer 502.

[0090]The audio encoder 509 is outputted to the buffer 504 for postrecording an AAU cycle. With this output, the corresponding PRU block in the buffer 504 for postrecording is overwritten. Record of postrecording data is performed by recording a PRU block on a predetermined ECC block.

[0091]Assuming sending Movie fragment atom and a PRU block to the buffer 504 for postrecording from the track buffer 502 in this model, It is for [of the ECC block accompanying rewriting of Movie fragment atom and PRU] omitting read-out again.

[0092]The seamless reproduction in this model shall be guaranteed if at least one-piece VU exists on the track buffer 502 at the time of the decoding start of VU. Output speed of data is set to R_s from ECC decoders 501. The maximum term by access which is read and record stops is set to T_a .

[0093]Seek time, latency speed, and time until the data first read from the disk after access is outputted from ECC are contained in these periods. In this example, it may be $R_s=20\text{Mbps}$ and $T_a=1\text{ second}$.

[0094]The reference postrecording algorithm in this example is explained using drawing 29. The numbers from (1) to (8) in drawing 29 are equivalent to the numbers from (1) to (8) under following explanation. The outline of an algorithm is as follows.

[0095](1) Read the data for reproduction. (2) Perform access to Movie fragment atom#N-1 at the same time encoding of the audio information corresponding to PRU#N which is Nth PRU is completed. (3) Record the PRU block corresponding to Movie fragment atom#N-1 and PRU#N on a disk. (4) Return to the original reading position. (5) Read the data for reproduction. Although the jump of division occurs ranging over RU boundary in the case of read-out, read-out is continued as it is.

[0096](6) Perform access to Movie fragment atom#N at the same time encoding of the audio information corresponding to PRU#N+1 which is the N+1st PRU(s) is completed. (7) Record the PRU block corresponding to Moviefragmentatom#N and PRU#N+1 on a disk. (8) Return to the original reading position. The above operation is

repeated.

[0097]In said reference device model, if the following conditions are fulfilled when it postrecords using said reference postrecording algorithm, it can guarantee that there is no underflow of the track buffer 502.

[0098]When maximum recording time of PRU in Tr(i) and RU#i is made into Tw(i) for the maximum reading time which includes T(i) and a division jump for the maximum reproducing time about RU#i in which the condition is arbitrary RUs in an AV stream, $Te(i) \geq Tr(i) + Tw(i)$... It is that <formula 1> is materialized.

[0099]Because, [0100] in arbitrary n in which this formula is a sufficient condition of seamless reproduction

[Mathematical formula 1]

$$\sum_{i=1}^n Te(i) \geq \sum_{i=1}^n (Tr(i) + Tw(i)) \quad \cdot \cdot \cdot < \text{式 } 2 >$$

[0101]It is because it is a ***** sufficient condition.

[0102]Since recording on the disk of postrecording data is performed synchronizing with the completion of PRU encoding, the data in the buffer 504 for postrecording does not accumulate, and there is also no overflow of the buffer 504 for postrecording.

[0103]When Tr(i) in <the formula 1> sets Ra, Rp, Rv, and ECC block size to Ly for the maximum bit rate of the main audio in an AV stream, a subaudio, and video, respectively, $Tr(i) = Te(i) \times (Rv + Ra + Rp) / Rs + Ta + Ly / Rs$... It is set to <formula 3>.

[0104]The 1st clause of the right-hand side expresses the reading time of RU#i. The 2nd clause of the right-hand side expresses the maximum access time by the division jump generated immediately after RU#i read-out. The 3rd clause of the right-hand side expresses Movie fragment atom reading time.

[0105]Although the size of Movie fragment atom is in RU reproducing time and proportionality, In about 10 seconds corresponding to the power off at the time of recording, in order that the size might not exceed 1 ECC block, reading time of Movie fragment atom was made into 1 ECC-block reading time.

[0106]Tw(i) is $Tw(i) = 2 Ta + Te(i) \times Rp / Rs + Ly / Rs + Ly / Rs$... It is <formula 4>.

[0107]Here, the 1st clause of the right-hand side shows the both-way access time to RU. Using maximum-access-time Ta for the access time of a round trip of PRU is based on the following Reasons.

[0108]It depends on the time delay by the buffer for reproduction at that time for the distance of the track read now and the track with which PRU which should be recorded exists. However, a time delay changes with buffer size for reproduction, and even if it is the same buffer size, also when read-out stops temporarily by a shock immediately before, they differ.

[0109]That is, the distance to access is unfixed, therefore it is necessary to estimate it with the worst value. The 2nd clause of the right-hand side expresses the sum total

of the time for recording PRU contained in RU#i on a disk. The 3rd clause of the right-hand side expresses the maximum of the record time other than the postrecording data in the ECC block in which PRU both ends are included. Since not the both ends of PRU are necessarily in agreement with an ECC block boundary, the Reason which needs such a clause is for recording mostly by a maximum of 1 ECC block from the size of PRU at the time of PRU record. The 4th clause of the right-hand side expresses the time which record of Movie fragment atom takes.

[0110]When <the formula 3> and the <formula 4> are substituted for <the formula 1> and it solves by $T_e(i)$ at this time, it is condition $T_e(i) \geq (3 \text{ TaxRs} + 3\text{Ly}) / (\text{Rs} - \text{Rv} - \text{Ra} - 2\text{Rp})$ of $T_e(i)$ which can guarantee real-time postrecording... <Formula 5> is obtained.

[0111]That is, RU reproducing time lower limit T_{emin} in which a postrecording guarantee is possible is $T_{\text{emin}} = (3 \text{ TaxRs} + 3\text{Ly}) / (\text{Rs} - \text{Rv} - \text{Ra} - 2\text{Rp})$... It is set to <formula 6>.

[0112]At this time, the upper limit T_{emax} of RU reproducing time is set up as follows. $T_{\text{emax}} = T_{\text{emin}} + T_{\text{vmax}}$... <formula 7> -- here, T_{vmax} is the maximum reproducing time of VU. Setting up upper limit is based on the following Reason. At the time of postrecording, the periods from (2) to (8) of drawing 29 become long as T_e becomes large. Since read-out from the disk of the data for playback cannot be performed, in order to continue playback, according to the increase in T_e , it is necessary to increase the size of the track buffer 502 in the meantime.

[0113]Upper limit is set up in order to enable the estimate of the size of the track buffer 502 which is needed at this time. When the margin for the maximum reproducing time of VU is between a lower limit and upper limit, it is possible to constitute RU from combination of arbitrary reproducing time.

[0114]Although the maximum reproducing time is set up here according to the bit rate of an AV stream, based on the greatest possible bit rate, it is not concerned with the bit rate of an AV stream, but is good also as fixed.

[0115]Processing when recording is directed from a <processing at time of record> user is explained with drawing 30. The AV streams recorded at this time are bit rate $\text{Rv} = 5\text{Mbps}$ of video, bit rate $\text{Ra} = 256\text{k bps}$ of a main audio, and bit rate $\text{Rp} = 256\text{k bps}$ of a postrecording audio, and are taken as the stream corresponding to fixed postrecording for VU reproducing time $T_v =$ about 0.5 second. The management information of a file system shall already be read on RAM.

[0116]First, it opts for the composition of a stream, or the composition of a continuation field (Step 701). When 1 VU is constituted from 15 1GOP, $\text{Rv} = 5\text{Mbps}$, $\text{Ra} = 256\text{k bps}$, $\text{Rp} = 256\text{k bps}$, and $T_{\text{vmax}} =$ about 0.5 second are substituted for <the formula 6 <formula 7>> for $\text{Rs} = 20\text{Mbps}$, and $T_a = 1$ second, and less than the more than range 4.4 second 4.9 second of $T_e(i)$ is obtained.

[0117]It is $T_e(i) = 4.5$ to fulfill this condition in $T_{\text{vmax}} =$ about 0.5 second. It becomes

the time of a second and PRU will be inserted for every nine VU. In MPEG-1 audio layer-II, at the time of bit rate 256k bps, the reproducing time T_{af} of AAU will be 0.024 second, size is set to 768 bytes, and the area size of PRU at this time is set to 144384 bytes. In a continuation field, one PRU, nine VU, and one Movie fragment atom are made to be contained.

[0118]First, continuously recordable free space is looked for for nine VU and one PRU. The Reason for not taking Movie fragment atom into consideration is because it is not necessary to record only top RU continuously with Movie fragment atom. Specifically, $9 \times T_{vmax} \times (R_a + R_v) + 9 \times T_{av} \times R_a$, i.e., the continuous free space of 24.8 or more Mbit, is looked for with reference to Space Bitmap on RAM102. If it does not exist, a user is told about the ability to stop recording and not record it (Step 702).

[0119]The audio encoder 117 and the video encoder 118 are started, respectively (Step 703). And it confirms whether the above data is stored in the buffer for record by 1 ECC block (32 KB) (Step 704), and while being accumulated, Step 708 is repeated from Step 705.

[0120]If accumulated, the empty situations of the ECC block on the disk recorded on the next will be investigated with reference to Space Bitmap on RAM (Step 705). If an opening becomes, the continuous free space which can record nine VU, PRU, and one Movie fragment atom will be looked for (Step 707), and a pickup will be moved to the head of the free space (Step 708).

[0121]And the data for 1 ECC block in the buffer 111 for record is recorded on a disk (Step 706). If data is not stored in the buffer 111 for record, it confirms whether the end of record is directed (Step 709), and Step 704 will be performed if it is not the end of record.

[0122]The following steps are performed when the end of record is directed. First, about the data which is less than 32 KB in the buffer for record, dummy data is added to an end and it is made 32 KB (Step 710). Next, the data is recorded on a disk (Step 711 - Step 714). The QuickTime management information and file system management information on RAM102 are recorded on the optical disc 106 (Step 715 - Step 716).

[0123]Operation of the audio encoder 117 and the video encoder 118 which are parallel with the above processing, or the multiplexer 113 is explained. Each encoder sends an encoding result to the multiplexer 113, and a multiplexer stores them in the buffer 114 for multiplexing.

[0124]If the data for 1 VU, i.e., 1GOP, and AAU reproduced synchronizing with it are accumulated in the buffer 114 for multiplexing, the multiplexer 113 will send 1-VU data to the buffer 111 for record. If that VU is the $9x_i$ -th VU (i is zero or more integers) at this time, PRU with above-mentioned size will be previously sent to the buffer 111 for record.

[0125]Reporting that the data for 1 VU has been encoded to the host CPU 101, the

host CPU 101 updates the QuickTime management information on RAM102 based on the number and size of GOP or AAU which constitute VU.

Processing when postrecording is directed from a <processing at time of postrecording> user is explained with drawing 31. Here, the QuickTime management information about the AV stream which is the target of postrecording shall already be read into RAM102.

[0126]First, the data for playback is read from the head of VU on a disk including a postrecording starting position (Step 802). At this time, Step 802 is repeated until it reads the data for sufficient reproducing time (Step 803). That it is enough here means only the data volume in which reproduction does not break off, even when the interruption period of the data read for reproduction is the maximum.

[0127]When PRU is read, the ECC block containing Movie fragment atom and PRU is sent to the buffer 111 for postrecording. In order to manage PRU in the buffer 111 for postrecording at this time, it holds to RAM102 by using the group of the reproduction time of onset (relative time from the head of an AV stream) of each PRU in the buffer 111 for postrecording, and the address in the inside of the buffer 111 for postrecording as a table.

[0128]Next, the video decoder 116, the audio decoder 115, and the audio encoder 117 are started (Step 804). The audio encoder 117 encodes the sampled voice waveform to AAU, and sends it to the multiplexer 113 with the cycle of AAU. In that case, the relative time from the head of an AV stream is added about each AAU.

[0129]The multiplexer 113 stores AAU in PRU in the buffer 111 for postrecording based on the time added to AAU. The contents of the postrecording audio track in Movie fragment atom read at Step 802 are updated, and it stores in the buffer 111 for postrecording. If it finishes storing AAU in PRU of the last in RU to the last, the end of encoding of RU will be notified to the host CPU 101.

[0130]Next, it is confirmed whether the end of postrecording is directed from the user (Step 805). The data for reproduction is read like Step 802 until encoding of PRU is completed, if not directed (Step 809).

[0131]When the end of RU encoding is notified from a multiplexer part (Step 806), From the playback time of onset of PRU contained in the RU currently held on the table on RAM102, it asks for the address on the optical disc 106 which should record the PRU, i.e., the address with which the PRU was recorded from the first, using QuickTime management information.

[0132]Except [record / of RU which started postrecording / PRU] the time, it asks for the address of last Movie fragment atom by lengthening for 1ECC from the address of calculated PRU. The pickup 107 is moved to the address (Step 807), and the ECC block containing Movie fragment atom and PRU is recorded on the optical disc 107 (Step 808).

[0133]If the end of postrecording is directed, it will wait for the completion of encoding

of PRU under present encoding (Step 810), A pickup is moved in quest of the recording address of Movie fragment atom in front of the PRU (Step 811), and Movie fragment atom and PRU are recorded (Step 812). Finally QuickTime management information is recorded on a disk (Step 813).

[0134]In this example, although the postrecorded data in PRU is managed by Movie fragment atom, carrying out by Movie atom#2 of the backup management information file 2402 is also considered. That is, in the backup management information file 2402, only a postrecording audio track will be managed by usual Sample table atom.

[0135]Also in this composition, the same contents as the management information of the postrecorded data of the movie file 2401, Since it exists in the backup management information file 2402, also when the movie file 2401 is damaged, it is possible to get to know whether it is finishing [postrecording of which field] by referring to the backup management information file 2402.

[0136]Since the management information of postrecorded data collects and exists in Movie atom of the backup management information file 2402, the renewal of management information is possible in a short time. However, in this composition, since the management information of the postrecorded data in PRU and PRU will exist in the position which separated on the disk, it becomes difficult to update management information at the time of postrecording, and it cannot respond to powering off at the time of postrecording like this example.

[0137]However, there are the following merits compared with this example. That is, it becomes possible by taking such composition to set up independently the insertion interval of Movie fragment atom at the time of postrecording, and the insertion interval of PRU. For example, it becomes possible to make powerful user-datum protection to powering off at the time of first time video record by inserting Movie fragment atom at an interval shorter than the insertion interval of PRU.

[0138]In order to fix fragmentation, when relocating data on a disk is taken into consideration, Although it is necessary to maintain that Record Unit in the movie file 2401 and Movie fragment atom of the backup management information file 2402 are continuation on a disk in this example, In such composition, the movie file 2401 should care only about what is continuously recorded per Record Unit, and does not need to care about the backup management information file 2402.

[0139]

[Effect of the Invention]As explained above, according to this invention, it becomes easy by arranging the field for postrecording, and backup management information continuously on a disk to update backup management information at the time of postrecording.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1] It is a block diagram showing the outline composition in the embodiment of this invention.

[Drawing 2] It is an explanatory view showing the relation of the management information and the AV stream in a QuickTime file format.

[Drawing 3] It is an explanatory view showing the outline of Movie atom in a QuickTime file format.

[Drawing 4] It is an explanatory view showing the outline of Track atom in a QuickTime file format.

[Drawing 5] It is an explanatory view showing the composition of Track header atom in a QuickTime file format.

[Drawing 6] It is an explanatory view showing the composition of Media atom in a QuickTime file format.

[Drawing 7] It is an explanatory view showing the composition of Media information atom in a QuickTime file format.

[Drawing 8] It is an explanatory view showing the example of the data management by Sample table atom.

[Drawing 9] It is an explanatory view showing the composition of Sample table atom in a QuickTime file format.

[Drawing 10] It is an explanatory view showing the composition of Edit atom in a QuickTime file format.

[Drawing 11] It is an explanatory view showing the example of the reproduction range specification by Edit atom.

[Drawing 12] It is an explanatory view showing the composition of User data atom in a QuickTime file format.

[Drawing 13] It is an explanatory view showing the entire configuration of Fragmented movie in a QuickTime file format.

[Drawing 14] It is an explanatory view showing the composition of Movie extends atom in a QuickTime file format.

[Drawing 15] It is an explanatory view showing the composition of Track extends atom in a QuickTime file format.

[Drawing 16] It is an explanatory view showing the composition of Movie fragment atom in a QuickTime file format.

[Drawing 17] It is an explanatory view showing the composition of Movie fragment header atom in a QuickTime file format.

[Drawing 18] It is an explanatory view showing the composition of Track fragment atom in a QuickTime file format.

[Drawing 19] It is an explanatory view showing the composition of Track fragment

header atom in a QuickTime file format.

[Drawing 20]It is an explanatory view showing the composition of Track fragment run atom in a QuickTime file format.

[Drawing 21]It is an explanatory view showing the composition of the stream corresponding to postrecording in one working example of this invention.

[Drawing 22]It is an explanatory view showing the structure of VU in one working example of this invention.

[Drawing 23]It is an explanatory view showing the relation between the movie file in one working example of this invention, and backup management information.

[Drawing 24]It is an explanatory view showing the continuous recording restrictions in one working example of this invention.

[Drawing 25]It is an explanatory view showing the management method of the postrecording field using the backup management information in one working example of this invention.

[Drawing 26]It is an explanatory view showing the example of Movie data of the backup management information in one working example of this invention.

[Drawing 27]It is an explanatory view showing the example of Movie fragment of the backup management information in one working example of this invention.

[Drawing 28]It is an explanatory view showing the reference device model in one working example of this invention.

[Drawing 29]It is an explanatory view showing the reference postrecording algorithm in one working example of this invention.

[Drawing 30]It is a flow chart which shows the recording operation in one working example of this invention.

[Drawing 31]It is a flow chart which shows the postrecording operation in one working example of this invention.

[Drawing 32]It is an explanatory view showing arrangement with the management information and the AV stream in conventional technology.

[Drawing 33]It is an explanatory view showing the AV stream corresponding to postrecording in conventional technology.

[Explanations of letters or numerals]

100 Bus

101 Host CPU

102 RAM

103 ROM

104 User interface

105 System clock

106 Optical disc

107 Pickup

108 ECC decoders

- 109 ECC encoder
- 110 The buffer for reproduction
- 111 The buffer for record/postrecording
- 112 Demultiplexer
- 113 Multiplexer
- 114 The buffer for multiplexing
- 115 Audio decoder
- 116 Video decoder
- 117 Audio encoder
- 118 Video encoder

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2003-22621
(P2003-22621A)

(43) 公開日 平成15年1月24日 (2003.1.24)

(51) Int.Cl. ⁷	識別記号	F I	テマコード* (参考)
G 1 1 B 20/12	1 0 3	G 1 1 B 20/12	5 C 0 5 3
20/10		20/10	1 0 3 5 D 0 4 4
27/00		27/00	G 5 D 1 1 0
H 0 4 N 5/92		H 0 4 N 5/92	A
			H
審査請求 未請求 請求項の数 8 O L (全 24 頁)			

(21) 出願番号 特願2001-207244(P2001-207244)

(22) 出願日 平成13年7月9日 (2001.7.9)

(71) 出願人 000005049

シャープ株式会社

大阪府大阪市阿倍野区長池町22番22号

(72) 発明者 木山 次郎

大阪府大阪市阿倍野区長池町22番22号 シ
ャープ株式会社内

(72) 発明者 岩野 裕利

大阪府大阪市阿倍野区長池町22番22号 シ
ャープ株式会社内

(74) 代理人 100102277

弁理士 佐々木 晴康 (外2名)

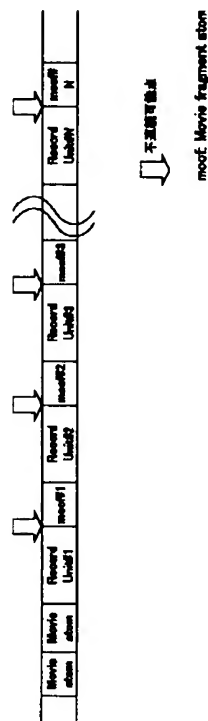
最終頁に続く

(54) 【発明の名称】 データ記録方法、データ変更方法及びその装置

(57) 【要約】

【課題】 AVストリームに多重化した管理情報の更新を容易にする。

【解決手段】 映像又は音声からなる第1のデータと、前記第1のデータと同期して再生される第2のデータとを格納する第1のユニット (Record Unit) と、前記第1のユニットを管理する管理情報 (Movie fragment atom) とを、記録媒体に記録するデータ記録方法であって、前記第1のユニットと前記管理情報とを、前記記録媒体上で近傍に配置するものである。



【特許請求の範囲】

【請求項 1】 映像又は音声からなる第 1 のデータと、前記第 1 のデータと同期して再生される第 2 のデータとを格納する第 1 のユニットと、前記第 1 のユニットを管理する管理情報とを、記録媒体に記録するデータ記録方法であって、

前記第 1 のユニットと前記管理情報とを前記記録媒体上で近傍に配置することを特徴とするデータ記録方法。

【請求項 2】 映像又は音声からなる第 1 のデータと、前記第 1 のデータと同期して再生される第 2 のデータとを格納する第 1 のユニットと、前記第 1 のデータを管理する第 1 の管理情報と、前記第 2 のデータを管理する第 2 の管理情報とを、記録媒体に記録するデータ記録方法であって、

前記第 2 の管理情報と前記第 1 の管理情報とを前記記録媒体上で分離して配置するとともに、1 個以上の前記第 2 の管理情報を互いに前記記録媒体上で近傍に配置することを特徴とするデータ記録方法。

【請求項 3】 映像又は音声からなる第 1 のデータと、前記第 1 のデータと同期して再生される第 2 のデータとを格納する第 1 のユニットと、前記第 1 のユニットを管理する管理情報とが、近傍に配置されて記録された記録媒体に対し、前記第 2 のデータと前記管理情報とを書き換えるデータ変更方法であって、前記第 2 のデータと前記管理情報とを連続的に書き換えることを特徴とするデータ変更方法。

【請求項 4】 映像又は音声からなる第 1 のデータと、前記第 1 のデータと同期して再生される第 2 のデータとを格納する第 1 のユニットと、前記第 1 のデータを管理する第 1 の管理情報とが記録され、さらに、前記第 2 の管理情報が 1 個以上互いに近傍に配置されて記録された記録媒体に対し、前記第 2 のデータと前記第 2 の管理情報とを書き換えるデータ変更方法であって、前記第 2 の管理情報を連続して書き換えることを特徴とするデータ変更方法。

【請求項 5】 映像又は音声からなる第 1 のデータと、前記第 1 のデータと同期して再生される第 2 のデータとを格納する第 1 のユニットと、前記第 1 のユニットを管理する管理情報とを、記録媒体に記録するデータ記録装置であって、前記管理情報を前記記録媒体上で前記第 1 のユニットの近傍に配置して記録する手段を備えたことを特徴とするデータ記録装置。

【請求項 6】 映像又は音声からなる第 1 のデータと、前記第 1 のデータと同期して再生される第 2 のデータとを格納する第 1 のユニットと、前記第 1 のデータを管理する第 1 の管理情報と、前記第 2 のデータを管理する第 2 の管理情報とを、記録媒体に記録するデータ記録装置であって、前記第 2 の管理情報と前記第 1 の管理情報とを前記記録

媒体上で分離して配置するとともに、1 個以上の前記第 2 の管理情報を互いに前記記録媒体上で近傍に配置して記録する手段を備えたことを特徴とするデータ記録装置。

【請求項 7】 映像又は音声からなる第 1 のデータと、前記第 1 のデータと同期して再生される第 2 のデータとを格納する第 1 のユニットと、前記第 1 のユニットを管理する管理情報とが、近傍に配置されて記録された記録媒体に対し、前記第 2 のデータと前記管理情報とを書き換えるデータ変更装置であって、前記管理情報を前記第 2 のデータと連続的に書き換える手段を備えたことを特徴とするデータ変更装置。

【請求項 8】 映像又は音声からなる第 1 のデータと、前記第 1 のデータと同期して再生される第 2 のデータとを格納する第 1 のユニットと、前記第 1 のデータを管理する第 1 の管理情報とが記録され、さらに、前記第 2 の管理情報が 1 個以上互いに近傍に配置されて記録された記録媒体に対し、前記第 2 のデータと前記第 2 の管理情報とを書き換えるデータ変更装置であって、前記第 2 の管理情報を連続して書き換える手段を備えたことを特徴とするデータ変更装置。

【発明の詳細な説明】**【0001】**

【発明の属する技術分野】本発明は、ハードディスク、光ディスク等のランダムアクセス可能な記録媒体に対して、映像データ、音声データを記録・変更するデータ記録方法、データ変更方法及びその装置に関するものである。

【0002】

【従来の技術】ディスクメディアを用いたビデオのデジタル記録再生装置が普及しつつある。ディスクメディアではテープメディアと異なり、任意の箇所へのアクセスが短時間で可能であり、その性質を利用して、データをコピーすることなく 1 枚のディスク上で管理情報の書換のみにより編集を行う非破壊編集あるいはノンリニア編集と呼ばれる機能や、記録済みのデータを上書きすることなく、削除した後の分散した空き領域にまたがるビデオを記録する機能を実現することができる。

【0003】ディスクメディアを用いたビデオ録画装置においては、録画時に管理情報を最後にまとめて記録することが一般的である。しかし、録画中に不意に電源が切れたりして管理情報を記録することができなかった場合、それまでに録画したデータを管理することができなくなってしまう、すなわち再生不能になってしまうという問題がある。

【0004】この問題に対応するため、例えば Motion JPEG 2000 では、Fragmented movie という概念が導入されている。ここでは、Fragmented movie についてその概略を説明する。Fragmented movie を含む Motion JPEG 2000 ファイルの典型的な構成を図 3 2 に示す。

【0005】先頭にそのファイル全体に共通する情報を管理するMovie atomが配置され、その後に、部分AVストリームデータ（Movie fragmentと呼ぶ）を格納するMovie data atomと、そのMovie fragmentを管理するMovie fragment atomとが交互に配置される。

【0006】録画時にはこの順番で記録を行なっていくことにより、仮に録画中に電源が切れた場合でも直前に記録したMovie fragment atomと、それが管理するMovie fragmentまではディスクに残っており、後で再生することが可能となる。

【0007】

【発明が解決しようとする課題】ディスクメディアにおいても、テープメディアと同様、アフレコ機能が要求される。アフレコ機能に対応したAVストリームとして、図33に示すように、アフレコ時にデータを格納するための領域を多重化したものが考えられる。

【0008】アフレコしたデータに関してもユーザの資産であり、管理情報のバックアップにより極力保護することが望まれる。しかしながら、上述のMotion JPEG2000規格においては、このようなストリームに関する管理情報の構成がない。

【0009】本発明は、上記課題に鑑みてなされたものであり、アフレコデータに関する管理情報の保護を可能にするデータ記録方法、データ変更方法を提供することを目的とする。

【0010】

【課題を解決するための手段】本願の第1の発明は、映像又は音声からなる第1のデータと、前記第1のデータと同期して再生される第2のデータとを格納する第1のユニットと、前記第1のユニットを管理する管理情報とを、記録媒体に記録するデータ記録方法であって、前記第1のユニットと前記管理情報とを前記記録媒体上で近傍に配置することを特徴とする。

【0011】本願の第2の発明は、映像又は音声からなる第1のデータと、前記第1のデータと同期して再生される第2のデータとを格納する第1のユニットと、前記第1のデータを管理する第1の管理情報と、前記第2のデータを管理する第2の管理情報とを、記録媒体に記録するデータ記録方法であって、前記第2の管理情報と前記第1の管理情報とを前記記録媒体上で分離して配置するとともに、1個以上の前記第2の管理情報を互いに前記記録媒体上で近傍に配置することを特徴とする。

【0012】本願の第3の発明は、映像又は音声からなる第1のデータと、前記第1のデータと同期して再生される第2のデータとを格納する第1のユニットと、前記第1のユニットを管理する管理情報とが、近傍に配置されて記録された記録媒体に対し、前記第2のデータと前記管理情報とを書き換えるデータ変更方法であって、前記第2のデータと前記管理情報とを連続的に書き換えることを特徴とする。

【0013】本願の第4の発明は、映像又は音声からなる第1のデータと、前記第1のデータと同期して再生される第2のデータとを格納する第1のユニットと、前記第1のデータを管理する第1の管理情報とが記録され、さらに、前記第2の管理情報が1個以上互いに近傍に配置されて記録された記録媒体に対し、前記第2のデータと前記第2の管理情報とを書き換えるデータ変更方法であって、前記第2の管理情報を連続して書き換えることを特徴とする。

【0014】本願の第5の発明は、映像又は音声からなる第1のデータと、前記第1のデータと同期して再生される第2のデータとを格納する第1のユニットと、前記第1のユニットを管理する管理情報とを、記録媒体に記録するデータ記録装置であって、前記管理情報を前記記録媒体上で前記第1のユニットの近傍に配置して記録する手段を備えたことを特徴とする。

【0015】本願の第6の発明は、映像又は音声からなる第1のデータと、前記第1のデータと同期して再生される第2のデータとを格納する第1のユニットと、前記第1のデータを管理する第1の管理情報と、前記第2のデータを管理する第2の管理情報とを、記録媒体に記録するデータ記録装置であって、前記第2の管理情報と前記第1の管理情報とを前記記録媒体上で分離して配置するとともに、1個以上の前記第2の管理情報を互いに前記記録媒体上で近傍に配置して記録する手段を備えたことを特徴とする。

【0016】本願の第7の発明は、映像又は音声からなる第1のデータと、前記第1のデータと同期して再生される第2のデータとを格納する第1のユニットと、前記第1のユニットを管理する管理情報とが、近傍に配置されて記録された記録媒体に対し、前記第2のデータと前記管理情報とを書き換えるデータ変更装置であって、前記管理情報を前記第2のデータと連続的に書き換える手段を備えたことを特徴とする。

【0017】本願の第8の発明は、映像又は音声からなる第1のデータと、前記第1のデータと同期して再生される第2のデータとを格納する第1のユニットと、前記第1のデータを管理する第1の管理情報とが記録され、さらに、前記第2の管理情報が1個以上互いに近傍に配置されて記録された記録媒体に対し、前記第2のデータと前記第2の管理情報とを書き換えるデータ変更装置であって、前記第2の管理情報を連続して書き換える手段を備えたことを特徴とする。

【0018】

【発明の実施の形態】以下、本発明の実施形態について、図面を参照しながら詳細に説明する。

【0019】＜システム構成＞図1は本実施形態において共通に用いる、アフレコ可能なビデオディスクレコーダの構成図である。この装置は、図1に示すように、バス100、ホストCPU101、RAM102、ROM103、ユーザインタ

フェース104、システムクロック105、光ディスク106、ピックアップ107、ECCデコーダ108、ECCエンコーダ109、再生用バッファ110、記録/アフレコ用バッファ111、デマルチプレクサ112、マルチプレクサ113、多重化用バッファ114、オーディオデコーダ115、ビデオデコーダ116、オーディオエンコーダ117、ビデオエンコーダ118、および図示しないカメラ、マイク、スピーカ、ディスプレイ等で構成される。

【0020】ホストCPU101は、バス100を通じてデマルチプレクサ112、マルチプレクサ113、ピックアップ107、また図示していないが、オーディオデコーダ115、ビデオデコーダ116、オーディオエンコーダ117、ビデオエンコーダ118との通信を行う。

【0021】再生時に、光ディスク106からピックアップ107を通じて読み出されたデータは、ECCデコーダ108によって誤り訂正され、再生用バッファ110に一旦蓄えられる。デマルチプレクサ112はオーディオデコーダ115、ビデオデコーダ116からのデータ送信要求に従い、再生用バッファ中のデータをその種別によって適当なデコーダに振り分ける。

【0022】一方、記録時に、オーディオエンコーダ117とビデオエンコーダ118によって圧縮符号化されたデータは、多重化用バッファ114に一旦送られ、マルチプレクサ113によってAV多重化され、記録/アフレコ用バッファ111に送られる。記録/アフレコ用バッファ111中のデータは、ECCエンコーダ109によって誤り訂正符号を付加され、ピックアップ107を通じて光ディスク106に記録される。

【0023】オーディオデータの符号化方式にはMPEG-1 Layer-IIを、ビデオデータの符号化方式にはMPEG-2をそれぞれ用いる。

【0024】光ディスク106は、外周から内周に向かって螺旋状に記録再生が行われる脱着可能な光ディスクとする。2048byteを1セクタとし、誤り訂正のため16セクタでECCブロックを構成する。ECCブロック中のデータを書き換える場合、そのデータが含まれるECCブロック全体を読み込み、誤り訂正を行って、対象のデータを書き換え、再び誤り訂正符号を付加し、ECCブロックを構成して、記録媒体に記録する必要がある。また、光ディスク106は、記録効率を上げるためZCAV（ゾーン角速度一定）を採用しており、記録領域は回転数の異なる複数のゾーンで構成される。

【0025】＜ファイルシステム＞光ディスク106上の各種情報を管理するためにファイルシステムを用いる。ファイルシステムには、パーソナルコンピュータ（PC）との相互運用を考慮してUDF（Universal Disk Format）を使用する。ファイルシステム上では、各種管理情報やAVストリームはファイルとして扱われる。

【0026】ユーザエリアは、2048byteの論理ブロック（セクタと一対一対応）で管理される。各ファイルは、

整数個のエクステント（連続した論理ブロック）で構成され、エクステント単位で分散して記録しても良い。空き領域は、Space Bitmapを用いて論理ブロック単位で管理される。

【0027】＜ファイルフォーマット＞AVストリーム管理のためのフォーマットとして、QuickTimeファイルフォーマットを用いる。QuickTimeファイルフォーマットとは、Apple社が開発したマルチメディアデータ管理用フォーマットであり、PCの世界で広く用いられている。

【0028】QuickTimeファイルフォーマットは、ビデオデータやオーディオデータ等（これらを総称してメディアデータとも呼ぶ）と管理情報とで構成される。両者を合わせてここでは、QuickTimeムービー（略してムービー）と呼ぶ。両者は同じファイル中に存在しても、別々のファイルに存在しても良い。

【0029】同じファイル中に存在する場合は、図2（a）に示すような構成をとる。各種情報はatomという共通の構造に格納される。管理情報はMovie atomという構造に格納され、AVストリームはMovie data atomという構造に格納される。尚、Movie atom中の管理情報には、メディアデータ中の任意の時間に対応するAVデータのファイル中での相対位置を導くためのテーブルや、メディアデータの属性情報や、後述する外部参照情報等が含まれている。

【0030】一方、管理情報とメディアデータを別々のファイルに格納した場合は、図2（b）に示すような構成をとる。管理情報はMovie atomという構造に格納されるが、AVストリームはatomには格納される必要はない。このとき、Movie atomはAVストリームを格納したファイルを「外部参照」している、という。

【0031】外部参照は、図2（c）に示すように、複数のAVストリームファイルに対して行うことが可能であり、この仕組みにより、AVストリーム自体を物理的に移動することなく、見かけ上編集を行ったように見せる、いわゆる「ノンリニア編集」「非破壊編集」が可能になる。

【0032】それでは、図3乃至図12を用いて、QuickTimeの管理情報のフォーマットについて説明する。まず、共通の情報格納フォーマットであるatomについて説明する。atomの先頭には、そのatomのサイズであるAtom size、そのatomの種別情報であるTypeが必ず存在する。Typeは4文字で区別され、例えばMovie atomでは'mov'、Movie data atomでは'mdat'となっている。

【0033】各atomは別のatomを含むことができる。すなわち、atom間には階層構造がある。Movie atomの構成を図3に示す。Movie header atomは、そのMovie atomが管理するムービーの全体的な属性を管理する。Track atomは、そのムービーに含まれるビデオやオーディオ等のトラックに関する情報を格納する。User data atomは、独自に定義可能なatomである。

【0034】Track atomの構成を図4に示す。Track header atomは、そのトラックの全体的な属性を管理する。Edit atomは、メディアデータのどの区間を、ムービーのどのタイミングで再生するかを管理する。Track reference atomは、別のトラックとの関係を管理する。Media atomは、実際のビデオやオーディオといったデータを管理する。

【0035】Track header atomの構成を図5に示す。ここでは、後での説明に必要なもののみについて説明する。flagsは属性を示すフラグの集合である。代表的なものとして、Track enabledフラグがあり、このフラグが1であれば、そのトラックは再生され、0であれば再生されない。layerはそのトラックの空間的な優先度を表しており、画像を表示するトラックが複数あれば、layerの値が小さいトラックほど画像が前面に表示される。

【0036】Media atomの構成を図6に示す。Media header atomは、そのMedia atomの管理するメディアデータに関する全体的な属性等を管理する。Handler reference atomは、メディアデータをどのデコーダでデコードするかを示す情報を格納する。Media information atomは、ビデオやオーディオ等メディア固有の属性情報を管理する。

【0037】Media information atomの構成を図7に示す。Media information header atomは、ビデオやオーディオ等メディア固有の属性情報を管理する。Handler reference atomは、Media atomの項で説明した通りである。Data information atomは、そのQuickTimeムービーが参照するメディアデータを含むファイルの名前を管理するatomであるData reference atomを含む。Sample table atomは、データのサイズや再生時間等を管理している。

【0038】次に、Sample table atomについて説明するが、その前に、QuickTimeにおけるデータの管理方法について、図8を用いて説明する。QuickTimeでは、データの最小単位（例えばビデオフレーム）をサンプルと呼ぶ。個々のトラック毎に、サンプルには再生時間順に1から番号（サンプル番号）がついている。

【0039】また、QuickTimeフォーマットでは、個々のサンプルの再生時間長およびデータサイズを管理している。また、同一トラックに属するサンプルが再生時間順にファイル中で連続的に配置された領域をチャンクと呼ぶ。チャンクにも再生時間順に、1から番号がついている。

【0040】さらに、QuickTimeフォーマットでは、個々のチャンクのファイル先頭からのアドレスおよび個々のチャンクが含むサンプル数を管理している。これらの情報に基づき、任意の時間に対応するサンプルの位置を求めることが可能となっている。

【0041】Sample table atomの構成を図9に示す。Sample description atomは、個々のチャンクのデータフ

ォーマット（Data format）やサンプルが格納されているファイルのチャンクのインデックス等を管理する。Time-to-sample atomは、個々のサンプルの再生時間を管理する。

【0042】Sync sample atomは、個々のサンプルのうち、デコード開始可能なサンプルを管理する。Sample-to-chunk atomは、個々のチャンクに含まれるサンプル数を管理する。Sample size atomは、個々のサンプルのサイズを管理する。Chunk offset atomは、個々のチャンクのファイル先頭からのアドレスを管理する。

【0043】Edit atomは、図10に示すように、1個のEdit list atomを含む。Edit list atomはNumber of entriesで指定される個数分の、Track duration、Media time、Media rateの値の組（エントリ）を持つ。各エントリは、トラック上で連続的に再生される区間に対応し、そのトラック上での再生時間順に並んでいる。

【0044】Track durationはそのエントリが管理する区間のトラック上での再生時間、Media timeはそのエントリが管理する区間の先頭に対応するメディアデータ上での位置、Media rateはそのエントリが管理する区間の再生スピードを表す。尚、Media timeが-1の場合は、そのエントリのTrack duration分、そのトラックでのサンプルの再生を停止する。この区間のことをempty editと呼ぶ。

【0045】図11にEdit listの使用例を示す。ここでは、Edit list atomの内容が図11（a）に示す内容であり、さらにサンプルの構成が図11（b）であったとする。尚、ここではi番目のエントリのTrack durationをD(i)、Media timeをT(i)、Media rateをR(i)とする。このとき、実際のサンプルの再生は、図11（c）に示す順に行われる。このことについて簡単に説明する。

【0046】まず、エントリ#1はTrack durationが13000、Media timeが20000、Media rateが1であるため、そのトラックの先頭から13000の区間はサンプル中の時刻20000から33000の区間を再生する。次に、エントリ#2はTrack durationが5000、Media timeが-1であるため、トラック中の時刻13000から18000の区間、何も再生を行わない。

【0047】最後に、エントリ#3はTrack durationが10000、Media timeが0、Media rateが1であるため、トラック中の時刻18000から28000の区間において、サンプル中の時刻0から10000の区間を再生する。

【0048】図12にUser data atomの構成を示す。このatomには、QuickTimeフォーマットで定義されていない独自の情報を任意個数格納することができる。1個の独自情報は1個のエントリで管理され、1個のエントリはSizeとTypeとUser dataで構成される。Sizeはそのエントリ自体のサイズを表し、Typeは独自情報をそれぞれ区別するための識別情報、User dataは実際のデータを表

す。

【0049】次に、録画中の電源遮断等に対応するために、QuickTimeファイルフォーマットに導入された概念であるFragmented Movieについて説明する。Fragmented movieは、QuickTime フォーマットの1アプリケーションであるMotion JPEG2000で導入された概念であり、上述のSample table atomに相当する情報を、部分的なAVストリーム毎に管理することが可能となっている。

【0050】Fragmented movieを導入したQuickTimeファイルの全体構成を図13に示す。先頭に、そのファイル全体に共通する情報を管理するMovie atomが配置され、その後、Movie fragmentを格納するMovie data atomと、そのMovie fragmentを構成するサンプルのアドレスやサイズ、再生時間等を管理するMovie fragment atomとが交互に配置される。尚、AVストリームデータは、通常のQuickTimeファイルと同様、別ファイルに存在しても構わない。

【0051】録画時にはこの順番で記録を行なっていくことにより、録画時の電源切断による被害を最小限に防ぐことが可能となっている。Movie atomには、そのQuickTimeムービーがFragmented movieであることを示すためのMovie extends atomが含まれる。Movie extends atomには、そのムービーに含まれる各トラックに関するデフォルト値が格納される。

【0052】また、Movie fragment atomには、そのMovie fragment atomが管理するMoviefragmentに関する管理情報が含まれている。管理情報には、その管理するMoviefragment全体に関する情報を格納するMovie fragment header atomと、Movie fragment中の各トラックに関する情報を格納するTrack fragment atomとがある。

【0053】Track fragment atomは、それが管理するトラックに属するMovie fragmentに関する情報を格納するTrack fragment header atomと、そのトラックに属するMovie fragmentを構成する論理的な連続領域(Track runと呼ばれる)をそれぞれ管理するTrack fragment run atomとを含む。以下に、各atomについて詳しく説明する。

【0054】Movie extends atomの構成を図14に示す。Movie extends atomは、前述のように、このatomを含むQuickTimeムービーがFragmented movieであることを示す役割を持つ。

【0055】Track extends atomの構成を図15に示す。Track extends atomは、このQuickTimeムービーに含まれる各トラックのサンプルのデフォルト値を設定するために存在する。track IDはMovie atom中で定義されているトラックのtrack IDを参照する。default-sample-で始まるフィールドは、このatomで管理されるtrack fragmentのデフォルト値を設定する。

【0056】Movie fragment atomの構成を図16に示す。録画中に逐次記録される管理情報はこのatomであ

る。このatomは、前述のとおり、このatomの管理するMovie fragmentに関する実際の情報を格納するatomであるMovie fragment header atomやTrack fragment atomを含む。

【0057】Movie fragment header atomの構成を図17に示す。このatomに格納されている主な情報はsequence-numberである。sequence-numberは、このatomが含まれるMovie fragment atomが管理するMovie fragmentの先頭からの順番を表す。

【0058】Track fragment atomの構成を図18に示す。Track fragment atomは、Moviefragmentに含まれる特定のトラックのサンプルに関する管理情報であるTrack fragment header atomやTrack fragment run atomを格納する。

【0059】Track fragment header atomの構成を図19に示す。このatomは、Movie fragmentに含まれる特定のトラックのサンプルに関するデフォルト値等を格納する。track-IDは、Movie atom中で定義されているトラックのtrack IDとの対応を示す。sample-description-indexは、このatomが管理するサンプルの参照するsample description tableのインデックス番号、default-sampleで始まるフィールドは、それぞれこのatomが管理するサンプルのデフォルト値である。

【0060】Track fragment run atomの構成を図20に示す。このatomは、Track runと呼ばれる、このatomの管理する連続領域や個々のサンプルの管理情報を格納する。sample-countはTrack runに含まれるサンプルの個数を示す。data-offsetはbase-data-offsetからのTrack runのオフセット値を示す。sample-で始まるフィールドはこのatomが管理するサンプルの再生時間等の値を格納する。ただし、上述のデフォルト値と同じであれば、省略してデータサイズを縮小することが可能となっている。

【0061】＜インデックス・ファイル＞ディスク内に含まれるQuickTimeムービーを管理するため、AVインデックス・ファイルという特別のQuickTimeムービーファイルをディスク内に1個置く。

【0062】AVインデックス・ファイルには、ディスク内のファイル(QuickTimeムービーやQuickTimeムービーから参照されている静止画等)に関するサムネイルや各種属性が登録されている。各種属性の中には、そのファイルが外部参照されている回数を示すlink countがある。

【0063】link countを参照することで、そのファイルを参照しているファイルがあるかどうかを容易に知ることができ、他から参照されているファイルを不用意に削除してしまうことを防ぐことができる。

【0064】＜実施例＞本発明の一実施例について、図21乃至図31を用いて説明する。

【0065】＜AVストリームの形態＞まず、本実施例に

おけるAVストリームの構成について、図21及び図22を用いて説明する。AVストリームは整数個のRecord Unit (RU) で構成される。RUはディスク上で連続的に記録する単位である。RUの長さは、AVストリームを構成するRUをどのようにディスク上に配置してもシームレス再生（再生中に画像や音声途切れなく再生できること）やリアルタイムアフレコ（アフレコ対象のビデオをシームレス再生しながらオーディオを記録すること）が保証されるように設定される。この設定方法については後述する。

【0066】また、RU境界がECCブロック境界と一致するようにストリームを構成する。RUのこれらの性質によって、AVストリームをディスクに記録した後も、シームレス再生を保証したまま、ディスク上でRU単位の配置を容易に変更することができる。

【0067】RUは、整数個のVideo Unit (VU) とPost Recording Unit (PRU) で構成される。VUは単独再生可能な単位であり、そのことから再生の際のエントリ・ポイントとなりうる。PRUは同じRUに含まれるVUと同期再生するデータを記録するための領域である。

【0068】VU構成を図22に示す。VUは、1秒程度のビデオデータを格納した整数個のGOP（グループ・オブ・ピクチャ）と、それらと同じ時間に再生されるメインオーディオデータを格納した整数個のAAU（オーディオ・アクセス・ユニット）とから構成される。

【0069】尚、GOPは、MPEG-2ビデオ規格における画像圧縮の単位であり、複数のビデオフレーム（典型的には15フレーム程度）で構成される。AAUはMPEG-1 Layer-II規格における音声圧縮の単位で、1152点の音波形サンプル点により構成される。サンプリング周波数が48kHzの場合、AAUあたりの再生時間は0.024秒となる。VU中では、AV同期再生のために必要となる遅延を小さくするため、AAU、GOPの順に配置する。

【0070】また、VU単位で独立再生を可能とするために、VU中のビデオデータの先頭にはSequence Header (SH) を、末尾にはSequence End Code (SEC) を置く。VUの再生時間は、VUに含まれるビデオフレーム数にビデオフレーム周期をかけたものと定義する。さらに、VUを整数個組み合わせてRUを構成する場合、RUの始末端をECCブロック境界に合わせるため、VUの末尾を0で埋める。

【0071】PRUの領域サイズは、ここではRUの再生時間にオーディオの最大ビットレートをかけたものとする。

【0072】＜AVストリーム管理方法＞上記のAVストリームの管理情報の構成について説明する。AVストリームは、図23に示すように、ムービーファイル2401と、ムービーファイル2401の管理情報のバックアップに相当する管理情報バックアップファイル2402とで管理される。

【0073】ムービーファイル2401は、Movie data ato

mに格納されたAVストリームと、AVストリームを構成するサンプルのアドレスやサイズ、再生時間等を管理するMovie atomとで構成される。AVストリームは、前述のように、RUで構成され、各RUは必ずディスク上で連続的に配置されるように記録される。

【0074】一方、管理情報バックアップファイル2402は、ムービーファイル2401中の各RUを管理するMovie fragment atomで構成される。この2つのファイルを、図23に示すように、ディスク上においてRU単位で多重化する。このような構成を取る理由を以下に説明する。

【0075】図13のような構成を取った場合、光ディスクにおいては録画時には都合が良いが再生時には都合が悪い。なぜならば、録画時にはシークを行うことなく、その時点までの管理情報をMovie fragment atomに記録できるのに対し、再生時には、Movie data atom#1を再生するためには、まずMovie fragment atom#1を読み込む必要があるため、ピックアップを行ったり来たりさせなければならない、読み込みの効率が悪くなるからである。

【0076】このとき、図23に示すような構成を取ることで、録画中の電源断やムービーファイル2401のMovie atomの破損時のバックアップが可能となると同時に、Movie fragment atomを用いない場合と同様、再生時の効率的な管理情報の読み込みが可能となる。

【0077】尚、上記の2ファイルはディスク上で多重化して記録するが、全体をディスク上で連続的に記録する必要はなく、各Record Unitと直前のMovie fragment とが連続的に記録されていれば良い。つまり、図24に示す不連続可能点以外ではディスク上で連続的に記録する。このような連続記録を行う理由は、アフレコ時においてPRUへの記録の直前に、アフレコに伴って必要となるMovie fragmentの更新を、シークが加わることなく可能にするためである。

【0078】また、図23においては、Movie fragment atomとRUとが接しているように記載されているが、実際にはMovie fragment atomとRUとの間には、Movie fragment atomの先頭からRUの直前までのバイト数が1ECCブロックサイズの整数倍になるように、パディングとしてFree space atomを挿入する。Free space atomとは、QuickTimeファイルフォーマットで規定されているatomで、領域の確保に利用される。

【0079】次に、ムービーファイル2401の管理情報の構成について説明する。ムービーファイル2401の管理情報は、ビデオデータ用のビデオトラック、メインオーディオ用のメインオーディオトラック、アフレコ領域管理用のアフレコ領域トラックおよび実際にアフレコされたデータを管理するためのアフレコオーディオトラックで構成される。

【0080】ビデオトラックは、Sequence headerからS

sequence end codeまでのビデオデータを1サンプル、VU中のビデオの塊を1チャンクとして管理する。メインオーディオトラックは、AAUを1サンプル、VU中のオーディオの塊を1チャンクとして管理する。アフレコ領域トラックは、PRUに所定のビットレートでオーディオデータを記録したものとして、AAUを1サンプル、PRUを1チャンクとして管理する。

【0081】ただし、このトラックはどのタイミングのオーディオデータをどこに記録するかを知らせるためのものであり、誤って再生しないよう、Track header atomのflagを再生不可に設定する。アフレコオーディオトラックは、PRUに実際に記録されているオーディオデータに関して1 AAUを1サンプル、PRU中の連続記録データを1チャンクとして管理する。

【0082】次に、管理情報バックアップファイル2402の管理情報の構成について説明する。ムービーファイル2401と同様、ビデオトラック、メインオーディオトラック、アフレコ領域トラック、アフレコオーディオトラックで構成し、AAUを1サンプル、Sequence headerからSequence end codeまでのビデオデータを1サンプルとして管理する。

【0083】バックアップ管理情報ファイルにおけるPRUの管理方法について、図25に示す構成のRUを例にとって説明する。ここでは、RUはムービーファイルの先頭からLバイトの位置にあるとする。また、PRUは、10個のAAUで構成され、先頭から4個のAAUは未アフレコ、後半の6個のAAUはアフレコ済みとする。そしてまた、各AAUは再生時間0.024秒、サイズは768バイトであるとする。

【0084】このとき、図26に示すように、Movie extends atom中のアフレコ領域トラックおよびアフレコオーディオトラックのdefault-sample-sizeは768、default-sample-durationは2160（1秒の細かさであるTimescaleを90000とした場合）となる。

【0085】また、このMovie fragmentを管理するMovie fragment atom中のアフレコ領域トラックおよびアフレコオーディオトラックの内容は、図27に示すとおりになる。まず、アフレコ領域トラックに関しては、PRU中の全サンプルを管理するため、sample-countは10となる。

【0086】次に、アフレコオーディオトラックに関しては、アフレコ済みのサンプルのみを管理するため、sample-countは6となり、data-offsetはPRUの先頭からアフレコ済みの先頭サンプルまでのバイト数となる。アフレコを行う際には、アフレコオーディオトラックのこれらの値を書き換えることになる。

【0087】<Record Unit再生時間決定方法>まず、アフレコ対応ストリームにおけるRU再生時間の決定方法について説明する。この決定方法では、機器間での互換性確保のため、基準となるデバイス（リファレンス・デ

バイス・モデル）と基準となるアフレコアルゴリズム（リファレンス・アフレコ・アルゴリズム）とを想定し、次にそれらを用いてアフレコを行った際にシームレス再生が破綻しないように、RU再生時間を決める。

【0088】それではまず、リファレンス・デバイス・モデルについて、図28を用いて説明する。リファレンス・デバイス・モデルは、1個のピックアップとそれにつながるECCエンコーダ・デコーダ501、トラックバッファ502、デマルチプレクサ503、アフレコ用バッファ504、オーディオエンコーダ509、ビデオバッファ505、オーディオバッファ506、ビデオデコーダ507、オーディオデコーダ508より構成される。

【0089】本モデルでは、ピックアップが1個であるため再生用データのディスクからの読み出しとアフレコ用データのディスクへの記録は時分割で行う。ディスクから再生用データを読み出す際、PRUと直前のMovie fragment atomを含めて読み出す。読み出されたMovie fragment atomとPRUを含むECCブロック（PRUブロック）は、トラックバッファ502からアフレコ用バッファ504に送られる。

【0090】オーディオエンコーダ509は、AAU周期でアフレコ用バッファ504に出力する。この出力によって、アフレコ用バッファ504中の対応するPRUブロックを上書きする。アフレコデータの記録は、PRUブロックを所定のECCブロックに記録することで行う。

【0091】本モデルにおいて、Movie fragment atomとPRUブロックをトラックバッファ502からアフレコ用バッファ504に送ることを想定しているのは、Movie fragment atomおよびPRUの書き換えに伴うECCブロックの再度読み出しを省略するためである。

【0092】本モデルにおけるシームレス再生は、VUのデコード開始時にトラックバッファ502上に少なくとも1個VUが存在すれば保証されるものとする。ECCデコーダ501からデータの出力速度はRsとする。また、アクセスによる読み出し、記録の停止する最大期間をTaとする。

【0093】尚、これら期間にはシーク時間、回転待ち時間、アクセス後に最初にディスクから読み出したデータがECCから出力されるまでの時間が含まれる。本実施例ではRs=20Mbps、Ta=1秒とする。

【0094】本実施例におけるリファレンス・アフレコ・アルゴリズムについて、図29を用いて説明する。尚、図29中の（1）から（8）までの番号は、以下の説明中の（1）から（8）までの番号に対応する。アルゴリズムの概要は次の通りである。

【0095】（1）再生用データの読み出しを行う。（2）N番目のPRUであるPRU#Nに対応するオーディオデータのエンコードが終了すると同時にMovie fragment atom#N-1へのアクセスを行う。（3）Movie fragment atom#N-1とPRU#Nに対応するPRUブロックをディスクに記録する。（4）元の読み出し位置に戻る。（5）再生用デ

ータの読み出しを行う。読み出しの際にRU境界を跨ぎ、分断のジャンプが発生するが、そのまま読み出しを続ける。

【0096】(6) N+1番目のPRUであるPRU#N+1に対応するオーディオデータのエンコードが終了すると同時に、Movie fragment atom#Nへのアクセスを行う。

(7) Movie fragment atom#NとPRU#N+1に対応するPRUブロックをディスクに記録する。(8) 元の読み出し位置に戻る。以上の動作を繰り返す。

【0097】前記リファレンス・デバイス・モデルにおいて、前記リファレンス・アフレコ・アルゴリズムを用いてアフレコを行った場合、次のような条件を満たせ

$$\sum_{i=1}^n Te(i) \geq \sum_{i=1}^n (Tr(i) + Tw(i)) \cdots \text{式 2}$$

【0101】を満たす十分条件であるためである。

【0102】また、PRUエンコード完了に同期してアフレコデータのディスクへの記録を行っているため、アフレコ用バッファ504中のデータが累積していくことはなく、アフレコ用バッファ504のオーバーフローもない。

$$Tr(i) = Te(i) \times (Rv + Ra + Rp) / (Rs + Ta + Ly / Rs) \cdots \text{式 3}$$

となる。

【0104】右辺第1項はRU#iの読み出し時間を表す。右辺第2項はRU#i読み出し直後に発生する分断ジャンプによる最大アクセス時間を表す。右辺第3項は、Movie fragment atom読み出し時間を表す。

【0105】尚、Movie fragment atomのサイズはRU再生時間と比例関係にあるが、録画時の電源断に対応した10秒程度ではそのサイズが1 ECCブロックを越えることはないため、Movie fragment atomの読み出し時間は1 ECCブロック読み出し時間とした。

【0106】また、Tw(i)は、
 $Tw(i) = 2Ta + Te(i) \times Rp / (Rs + Ly / Rs + Ly / Rs) \cdots \text{式 4}$
 である。

【0107】ここで、右辺第1項はRUへの往復アクセス時間を示す。PRUへの往復のアクセス時間に最大アクセス時間Taを用いているのは、以下の理由に基づく。

【0108】現在読み出ししているトラックと記録すべきPRUの存在するトラックの距離は、そのときの再生用バ

$$Te(i) \geq (3Ta \times Rs + 3Ly) / (Rs - Rv - Ra - 2Rp) \cdots \text{式 5}$$

が得られる。

【0111】つまり、アフレコ保証可能なRU再生時間下限値Teminは、
 $Temin = (3Ta \times Rs + 3Ly) / (Rs - Rv - Ra - 2Rp) \cdots \text{式 6}$
 となる。

【0112】このとき、RU再生時間の上限値Tmaxを次のように設定する。

$$Tmax = Temin + Tvmax \cdots \text{式 7}$$

ここで、TvmaxはVUの最大再生時間である。上限値を設定するのは、次の理由に基づく。Teが大きくなるにした

ば、トラックバッファ502のアンダーフローがないことを保証することができる。

【0098】その条件とは、AVストリーム中の任意のRUであるRU#iについて最大再生時間をT(i)、分断ジャンプを含めた最大読み出し時間をTr(i)、RU#i中のPRUの最大記録時間をTw(i)としたとき、

$$Te(i) \geq Tr(i) + Tw(i) \cdots \text{式 1}$$

が成立することである。

【0099】なぜなら、この式は、シームレス再生の十分条件である任意のnにおける

【0100】

【数1】

【0103】<式1>中のTr(i)は、AVストリーム中のメインオーディオとサブオーディオ、ビデオの最大ビットレートをそれぞれRa、Rp、Rv、ECCブロックサイズをLyとしたとき、

ッファによる遅延時間に依存する。しかし、遅延時間は再生用バッファサイズによって異なり、また同じバッファサイズであっても、直前に衝撃によって読み出しが一時的に停止した場合にも異なる。

【0109】すなわち、アクセスする距離は不定であり、そのため最悪値で見積もる必要がある。右辺第2項は、RU#iに含まれるPRUをディスクに記録するための時間の合計を表す。右辺第3項はPRU両端が含まれるECCブロック中のアフレコデータ以外の記録時間の最大値を表している。このような項が必要な理由は、PRUの両端はECCブロック境界と一致しているとは限らないため、PRU記録時には、PRUのサイズより最大1 ECCブロック分多く記録することになるためである。右辺第4項はMovie fragment atomの記録に要する時間を表す。

【0110】このとき、<式1>に<式3>と<式4>を代入して、Te(i)で解くと、リアルタイムアフレコを保証可能なTe(i)の条件

が、アフレコ時において図29の(2)から(8)までの期間が長くなる。この間は再生用データのディスクからの読み出しができないため、再生を継続するためには、Teの増加に応じて、トラックバッファ502のサイズを増やす必要がある。

【0113】上限値を設定するのは、このとき必要となるトラックバッファ502のサイズを見積り可能にするためである。また、下限値と上限値の間にVUの最大再生時間分のマージンがあることにより、任意の再生時間の組み合わせでRUを構成することが可能である。

【0114】尚、ここでは最大再生時間をAVストリームのビットレートに応じて設定しているが、可能な最大のビットレートに基づき、AVストリームのビットレートに関わらず一定としても良い。

【0115】＜記録時の処理＞ユーザから録画が指示された場合の処理を、図30とともに説明する。このとき記録するAVストリームは、ビデオのビットレート $R_v=5\text{Mbps}$ 、メインオーディオのビットレート $R_a=256\text{kbps}$ 、アフレコオーディオのビットレート $R_p=256\text{kbps}$ で、VU再生時間 $T_v\approx 0.5$ 秒固定アフレコ対応ストリームとする。また、ファイルシステムの管理情報はすでにRAM上に読み込まれているものとする。

【0116】まず、ストリームの構成や連続領域の構成を決定する(ステップ701)。1VUを1GOP15フレームで構成するとしたとき、＜式6＞＜式7＞に $R_s=20\text{Mbps}$ 、 $T_a=1$ 秒、 $R_v=5\text{Mbps}$ 、 $R_a=256\text{kbps}$ 、 $R_p=256\text{kbps}$ 、 $T_{vmax}\approx 0.5$ 秒を代入し、 $T_e(i)$ の範囲4.4秒以上4.9秒以下が得られる。

【0117】 $T_{vmax}\approx 0.5$ 秒でこの条件を満たすのは、 $T_e(i)=4.5$ 秒のときとなり、9個のVU毎にPRUが挿入されることになる。MPEG-1 audio layer-IIにおいて、ビットレート256kbpsのとき、AAUの再生時間 T_{af} は0.024秒、サイズは768byteとなり、このときのPRUの領域サイズは、144384byteとなる。また、連続領域には1個のPRUと9個のVUと1個のMovie fragment atomとが含まれるようにする。

【0118】まず、9個のVUと1個のPRUを連続的に記録可能な空き領域を探す。Movie fragment atomを考慮しない理由は、先頭のRUだけはMovie fragment atomと連続的に記録する必要がないためである。具体的には、 $9 \times T_{vmax} \times (R_a + R_v) + 9 \times T_{av} \times R_a$ 、つまり24.8Mbit以上の連続的な空き領域をRAM102上のSpace Bitmapを参照して探す。存在しなければ、録画を中止し、録画できないことをユーザに知らせる(ステップ702)。

【0119】また、オーディオエンコーダ117、ビデオエンコーダ118をそれぞれ起動する(ステップ703)。そして、記録用バッファに1ECCブロック分(32KB)以上のデータが蓄積されているかどうかをチェックし(ステップ704)、蓄積されている間、ステップ705からステップ708を繰り返す。

【0120】蓄積されていれば、次に記録するディスク上のECCブロックの空き状況をRAM上のSpace Bitmapを参照して調べる(ステップ705)。空きがなれば、9個のVUとPRUと1個のMovie fragment atomを記録可能な連続的な空き領域を探し(ステップ707)、その空き領域の先頭へピックアップを移動する(ステップ708)。

【0121】そして、記録用バッファ111中の1ECCブロック分のデータをディスクに記録する(ステップ706)。記録用バッファ111にデータが蓄積されていないければ、記録終了が指示されているかどうかをチェックし

(ステップ709)、記録終了でなければ、ステップ704を実行する。

【0122】記録終了が指示されていた場合、以下のステップを実行する。まず、記録用バッファ中の32KBに満たないデータに関して、末尾にダミーデータを付加し32KBにする(ステップ710)。次に、そのデータをディスク上に記録する(ステップ711～ステップ714)。RAM102上のQuickTime管理情報とファイルシステム管理情報とを光ディスク106に記録する(ステップ715～ステップ716)。

【0123】以上の処理と並行するオーディオエンコーダ117、ビデオエンコーダ118やマルチプレクサ113の動作について説明する。それぞれのエンコーダはマルチプレクサ113にエンコード結果を送り、マルチプレクサはそれらを多重化用バッファ114に格納する。

【0124】1VU分のデータ、つまり1GOPとそれに同期して再生されるAAUが多重化用バッファ114に蓄積されたら、マルチプレクサ113は記録用バッファ111に1VUのデータを送る。このとき、そのVUが $9 \times i$ 番目(i は0以上の整数)のVUであつたら、上述のサイズを持ったPRUを先に記録用バッファ111に送る。

【0125】さらに、ホストCPU101に1VU分のデータがエンコードできたことを通知し、ホストCPU101はVUを構成するGOPやAAUの数およびサイズを基にRAM102上のQuickTime管理情報を更新する。

＜アフレコ時の処理＞ユーザからアフレコが指示された場合の処理を図31とともに説明する。ここで、アフレコの対象となるAVストリームに関するQuickTime管理情報は、すでにRAM102に読み込まれているものとする。

【0126】まず、アフレコ開始位置を含むディスク上のVUの先頭から再生用データの読み出しを行う(ステップ802)。このとき、十分な再生時間分のデータを読み出すまでステップ802を繰り返す(ステップ803)。ここで十分とは、再生用データ読み出しの中断期間が最大の場合でも再生が途切れないだけのデータ量を意味する。

【0127】また、PRUを読み出した際には、Movie fragment atomおよびPRUを含むECCブロックをアフレコ用バッファ111に送る。このとき、アフレコ用バッファ111中のPRUを管理するために、アフレコ用バッファ111中の各PRUの再生開始時間(AVストリームの先頭からの相対時間)とアフレコ用バッファ111中でのアドレスの組をテーブルとしてRAM102に保持する。

【0128】次に、ビデオデコーダ116とオーディオデコーダ115および、オーディオエンコーダ117を起動する(ステップ804)。オーディオエンコーダ117は、サンプリングされた音声波形をAAUにエンコードし、AAUの周期でマルチプレクサ113に送る。その際に、各AAUについてAVストリームの先頭からの相対時間を付加する。

【0129】マルチプレクサ113は、AAUに付加された時間に基づき、AAUをアフレコ用バッファ111中のPRUに格

納する。また、ステップ802で読み出したMovie fragment atom中のアフレコオーディオトラックの内容を更新し、アフレコ用バッファ111に格納する。RU中の最後のPRUにAAUを最後まで格納し終わったら、ホストCPU101にRUのエンコード終了を通知する。

【0130】次に、ユーザからアフレコ終了を指示されていないかチェックする（ステップ805）。指示されていないければ、PRUのエンコードが終了するまで、ステップ802と同様に、再生用データの読み出しを行う（ステップ809）。

【0131】マルチプレクサ部からRUエンコード終了が通知されたら（ステップ806）、RAM102上のテーブルに保持しているそのRUに含まれるPRUの再生開始時間から、QuickTime管理情報を用いて、そのPRUを記録すべき光ディスク106上のアドレス、つまり元々そのPRUが記録されていたアドレスを求める。

【0132】アフレコを開始したRUのPRU記録の時以外は、求めたPRUのアドレスから1 ECC分の引くことで直前のMovie fragment atomのアドレスを求める。そのアドレスにピックアップ107を移動させ（ステップ807）、Movie fragment atomとPRUを含むECCブロックを光ディスク107に記録する（ステップ808）。

【0133】アフレコ終了が指示されていれば、現在エンコード中のPRUのエンコード完了を待って（ステップ810）、そのPRUの直前のMovie fragment atomの記録アドレスを求めピックアップを移動し（ステップ811）、Movie fragment atomとPRUを記録する（ステップ812）。最後にQuickTime管理情報をディスクに記録する（ステップ813）。

【0134】尚、本実施例においては、PRU中のアフレコ済みデータの管理を、Movie fragment atomで行っているが、バックアップ管理情報ファイル2402のMovie atom#2で行うことも考えられる。すなわち、バックアップ管理情報ファイル2402において、アフレコオーディオトラックのみ通常のSample table atomで管理することになる。

【0135】この構成においても、ムービーファイル2401のアフレコ済みデータの管理情報と同様の内容は、バックアップ管理情報ファイル2402に存在するため、ムービーファイル2401が破損した場合も、バックアップ管理情報ファイル2402を参照することで、どの領域がアフレコ済みかを知ることが可能である。

【0136】また、アフレコ済みデータの管理情報は、バックアップ管理情報ファイル2402のMovie atomにまとも存在するため、管理情報の更新は短時間で可能である。ただしこの構成では、PRUとPRU中のアフレコ済みデータの管理情報とがディスク上で離れた位置に存在することになるため、アフレコ時に管理情報を更新することは困難となり、本実施例のようにアフレコ時の電源切断に対応することはできない。

【0137】しかし、本実施例に比べて、次のようなメリットがある。すなわち、このような構成を取ることによって、アフレコ時のMovie fragment atomの挿入間隔とPRUの挿入間隔を独立に設定することが可能となる。例えば、Movie fragment atomをPRUの挿入間隔より短い間隔で挿入することで、初回ビデオ記録時の電源切断に対するユーザデータ保護を強力にすることが可能となる。

【0138】また、フラグメンテーションを解消するため、ディスク上でデータの再配置を行うことを考慮した場合、本実施例ではムービーファイル2401中のRecord Unitとバックアップ管理情報ファイル2402のMovie fragment atomとがディスク上で連続であることを保つ必要があるが、このような構成の場合、ムービーファイル2401がRecord Unit単位に連続的に記録されることだけを気にすればよく、バックアップ管理情報ファイル2402に関しては気にする必要がない。

【0139】

【発明の効果】以上説明したように、本発明によれば、アフレコ用領域とバックアップ管理情報をディスク上で連続的に配置することにより、アフレコ時においてバックアップ管理情報を更新することが容易となる。

【図面の簡単な説明】

【図1】本発明の実施形態における概略構成を示すブロック図である。

【図2】QuickTimeファイルフォーマットにおける管理情報とAVストリームとの関係を示す説明図である。

【図3】QuickTimeファイルフォーマットにおけるMovie atomの概要を示す説明図である。

【図4】QuickTimeファイルフォーマットにおけるTrack atomの概要を示す説明図である。

【図5】QuickTimeファイルフォーマットにおけるTrack header atomの構成を示す説明図である。

【図6】QuickTimeファイルフォーマットにおけるMedia atomの構成を示す説明図である。

【図7】QuickTimeファイルフォーマットにおけるMedia information atomの構成を示す説明図である。

【図8】Sample table atomによるデータ管理の例を示す説明図である。

【図9】QuickTimeファイルフォーマットにおけるSample table atomの構成を示す説明図である。

【図10】QuickTimeファイルフォーマットにおけるEdit atomの構成を示す説明図である。

【図11】Edit atomによる再生範囲指定の例を示す説明図である。

【図12】QuickTimeファイルフォーマットにおけるUser data atomの構成を示す説明図である。

【図13】QuickTimeファイルフォーマットにおけるFragmented movieの全体構成を示す説明図である。

【図14】QuickTimeファイルフォーマットにおけるMovie extends atomの構成を示す説明図である。

【図15】QuickTimeファイルフォーマットにおけるTrack extends atomの構成を示す説明図である。

【図16】QuickTimeファイルフォーマットにおけるMovie fragment atomの構成を示す説明図である。

【図17】QuickTimeファイルフォーマットにおけるMovie fragment header atomの構成を示す説明図である。

【図18】QuickTimeファイルフォーマットにおけるTrack fragment atomの構成を示す説明図である。

【図19】QuickTimeファイルフォーマットにおけるTrack fragment header atomの構成を示す説明図である。

【図20】QuickTimeファイルフォーマットにおけるTrack fragment run atomの構成を示す説明図である。

【図21】本発明の一実施例におけるアフレコ対応ストリームの構成を示す説明図である。

【図22】本発明の一実施例におけるVUの構造を示す説明図である。

【図23】本発明の一実施例におけるムービーファイルとバックアップ管理情報の関係を示す説明図である。

【図24】本発明の一実施例における連続記録制限を示す説明図である。

【図25】本発明の一実施例におけるバックアップ管理情報によるアフレコ領域の管理方法を示す説明図である。

【図26】本発明の一実施例におけるバックアップ管理情報のMovie dataの例を示す説明図である。

【図27】本発明の一実施例におけるバックアップ管理情報のMovie fragmentの例を示す説明図である。

【図28】本発明の一実施例におけるリファレンス・デバイス・モデルを示す説明図である。

【図29】本発明の一実施例におけるリファレンス・ア

フレコ・アルゴリズムを示す説明図である。

【図30】本発明の一実施例における記録動作を示すフローチャートである。

【図31】本発明の一実施例におけるアフレコ動作を示すフローチャートである。

【図32】従来技術における管理情報とAVストリームとの配置を示す説明図である。

【図33】従来技術におけるアフレコ対応AVストリームを示す説明図である。

【符号の説明】

- 100 バス
- 101 ホストCPU
- 102 RAM
- 103 ROM
- 104 ユーザインタフェース
- 105 システムクロック
- 106 光ディスク
- 107 ピックアップ
- 108 ECCデコーダ
- 109 ECCエンコーダ
- 110 再生用バッファ
- 111 記録/アフレコ用バッファ
- 112 デマルチプレクサ
- 113 マルチプレクサ
- 114 多重化用バッファ
- 115 オーディオデコーダ
- 116 ビデオデコーダ
- 117 オーディオエンコーダ
- 118 ビデオエンコーダ

【図3】

```
Movie atom {
    Atom size
    Type(='moov')
    Movie header atom
    Track atom (video track)
    Track atom (main audio track)
    :
    user data atom
}
```

【図4】

```
Track atom {
    Atom size
    Type(='trak')
    Track header atom
    Edit atom
    Track reference atom
    Media atom
    user data atom
    :
```

【図5】

```
Track header atom {
    Atom size
    Type(='tkhd')
    Version
    Flags
    Creation time
    Modification time
    Track ID
    Reserved
    Duration
    Reserved
    Layer
    Alternate group
    Volume
    Reserved
    Matrix structure
    Track width
    Track height
}
```

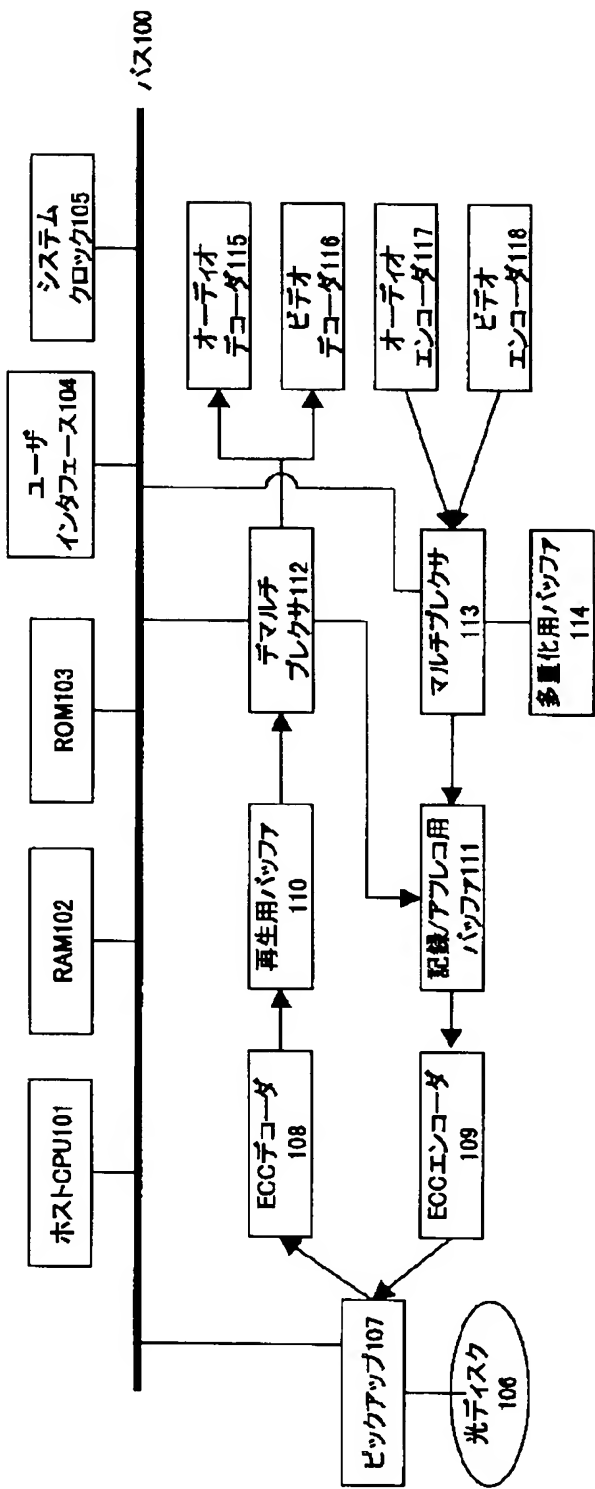
【図6】

```
Media atom {
    Atom size
    Type(='mdia')
    Media header atom
    Handler reference atom
    Media information atom
    User data atom
    :
```

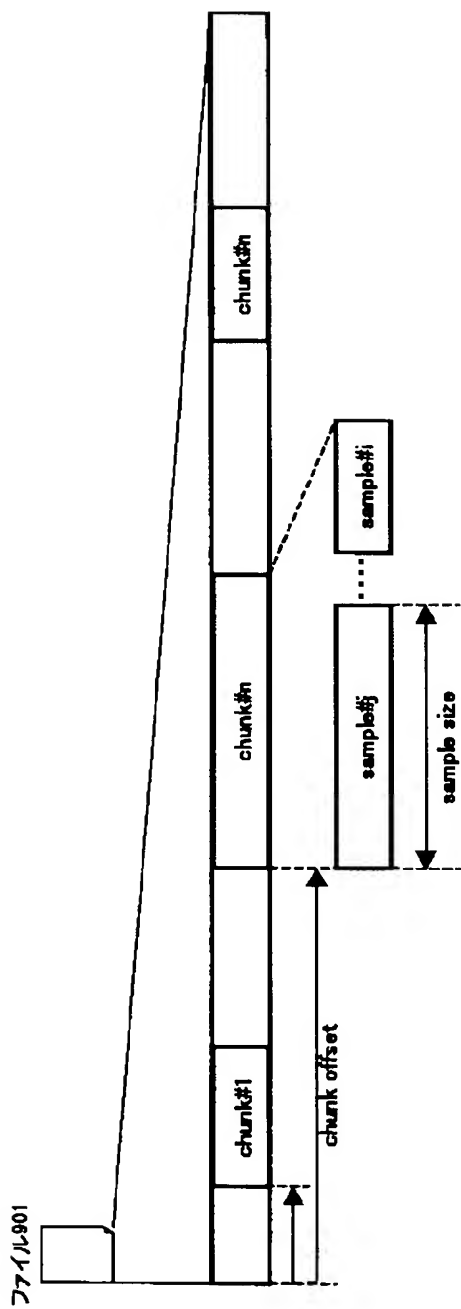
【図17】

```
Movie fragment header atom {
    Atom size
    Type(='mfhd')
    version
    flags
    sequence-number
}
```

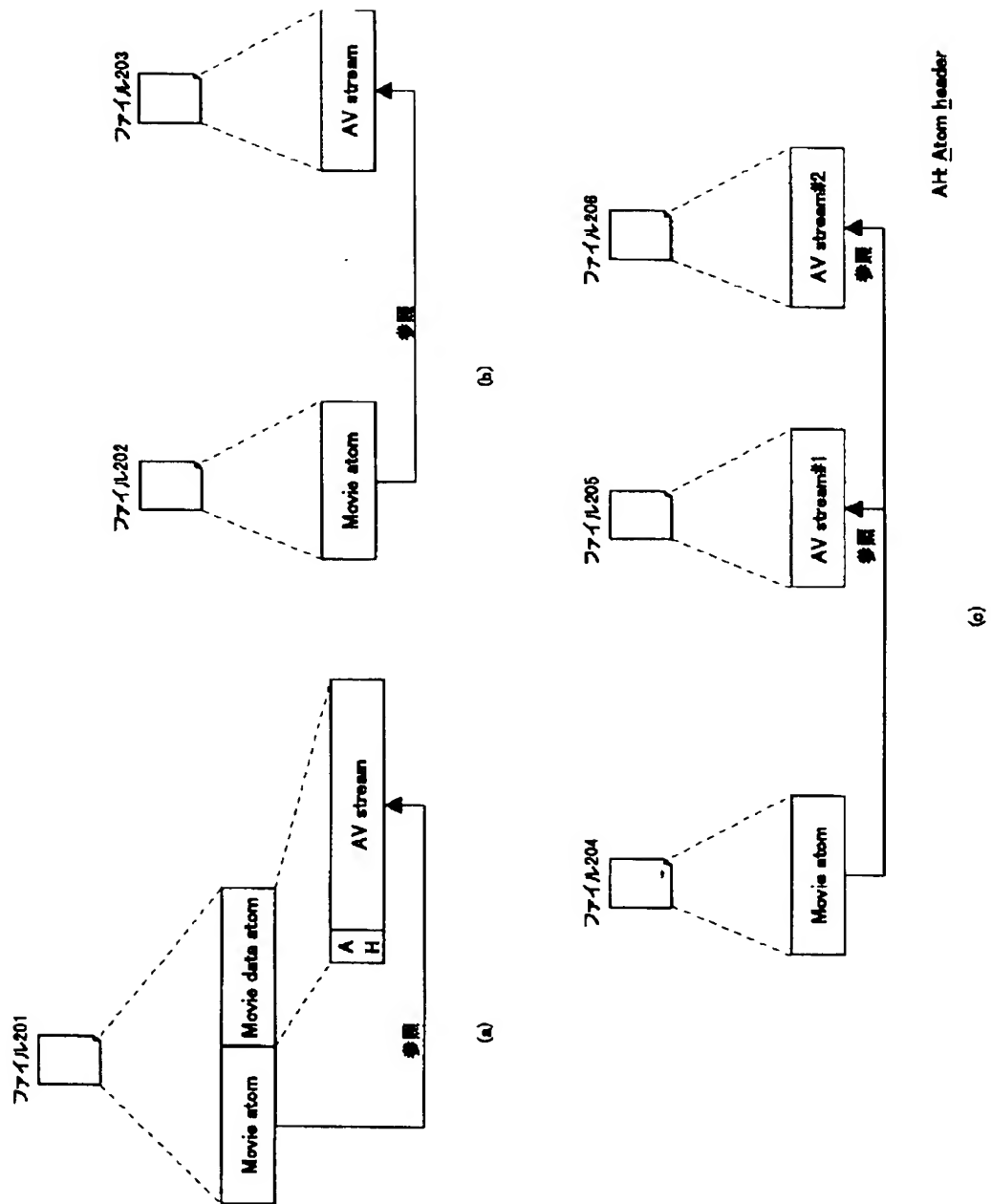
【図1】



【図8】



【図 2】



【図7】

```
Media information atom {
    Atom size
    Type(='minf')
    {video or Sound or Base} media information header atom
    Handler reference atom
    Data information atom
    Sample table atom
}
```

【図9】

```
Sample table atom {
    Atom size
    Type(='stbl')
    Sample description atom
    Time-to-sample atom
    Sync sample atom
    Sample-to-chunk atom
    Sample size atom
    Chunk offset atom
}
```

【図10】

【図12】

```
Edit atom {
    Atom size
    Type(='edts')
    Edit list atom
}
```

```
User data atom {
    Atom size
    Type(='udta')
    for (i=0; i<N; i++){
        Atom size
        Type
        User data
    }
}
```

【図14】

```
Movie extends atom {
    Atom size
    Type(='mvex')
    version
    flags
    for (i=0; i<n; i++){
        Track extends atom
    }
}
```

```
Edit list atom {
    Atom size
    Type(='elst')
    Versions
    Flags
    Number of entries(=N)
    for (i = 0; i < N; i++){
        Track duration
        Media time
        Media rate
    }
}
```

【図16】

【図15】

```
Track extends atom {
    Atom size
    Type(='trex')
    version
    flags
    track-ID
    default-sample-description-index
    default-sample-duration
    default-sample-size
    default-sample_flags
}
```

```
Movie fragment atom {
    Atom size
    Type(='moof')
    version
    flags
    Movie fragment header atom
    for (i=0; i<n; i++){
        Track fragment atom
    }
}
```

【図19】

【図18】

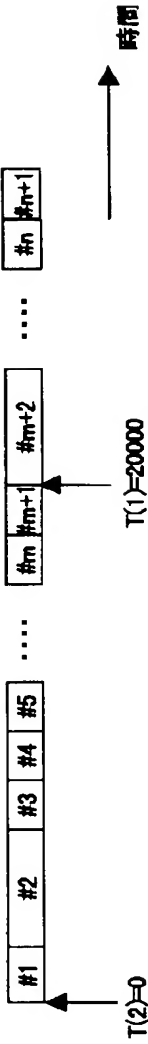
```
Track fragment atom {
    Atom size
    Type(='traf')
    version
    flags
    Track fragment header atom
    for (i=0; i<n; i++){
        Track fragment run atom
    }
}
```

```
Track fragment header atom {
    Atom size
    Type(='tfhd')
    version
    flags
    track-ID
    base-data-offset
    sample-description-index
    default-sample-duration
    default-sample-size
    default-sample_flags
}
```

(a)

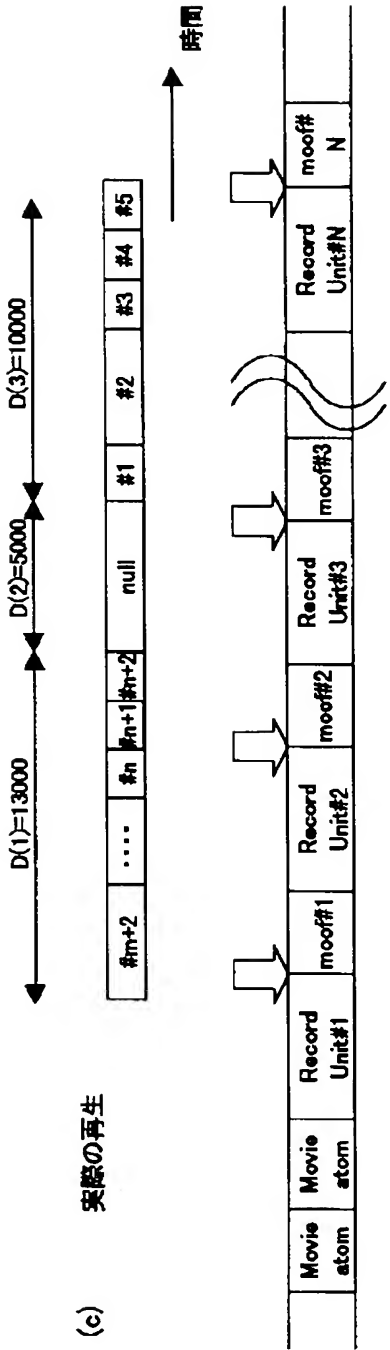
Entry Number	Track duration D(i)	Media time T(i)	Media rate R(i)
#1	13000	20000	1
#2	5000	-1	1
#3	10000	0	1

(b) Sample



【図 1 1】

(c) 実際の再生

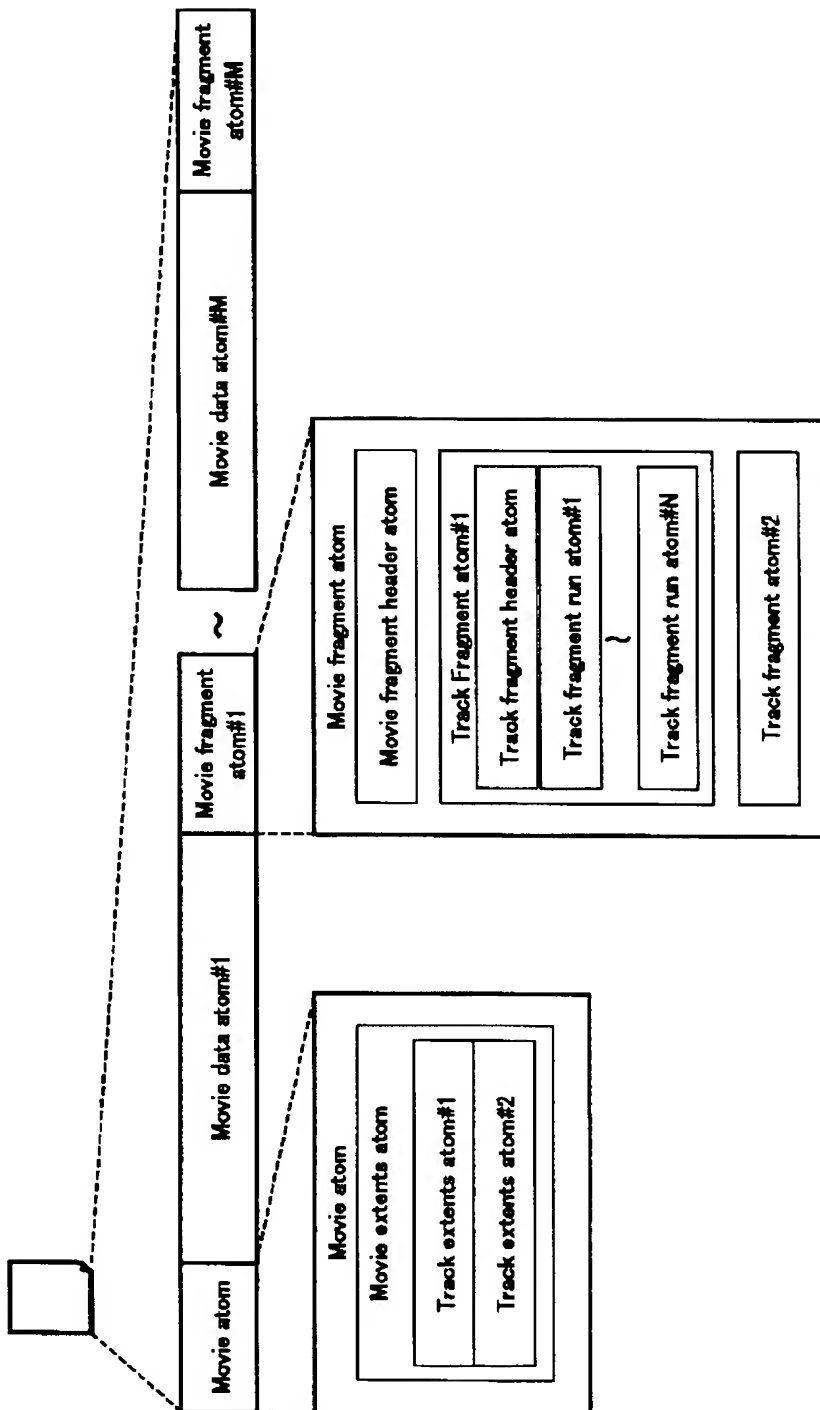


【図 2 4】

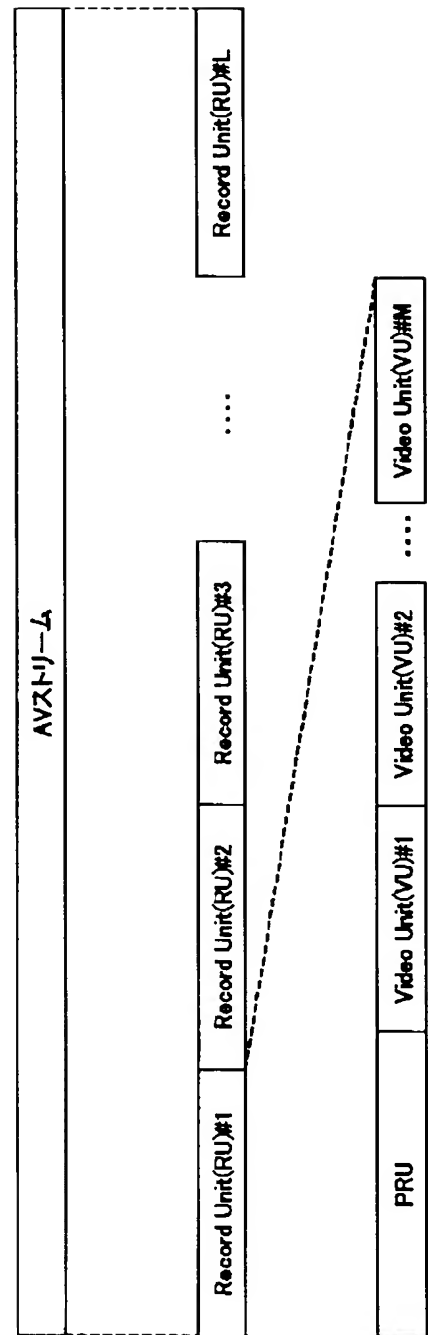
不連続可能点

moof: Movie fragment atom

【図13】



【図21】



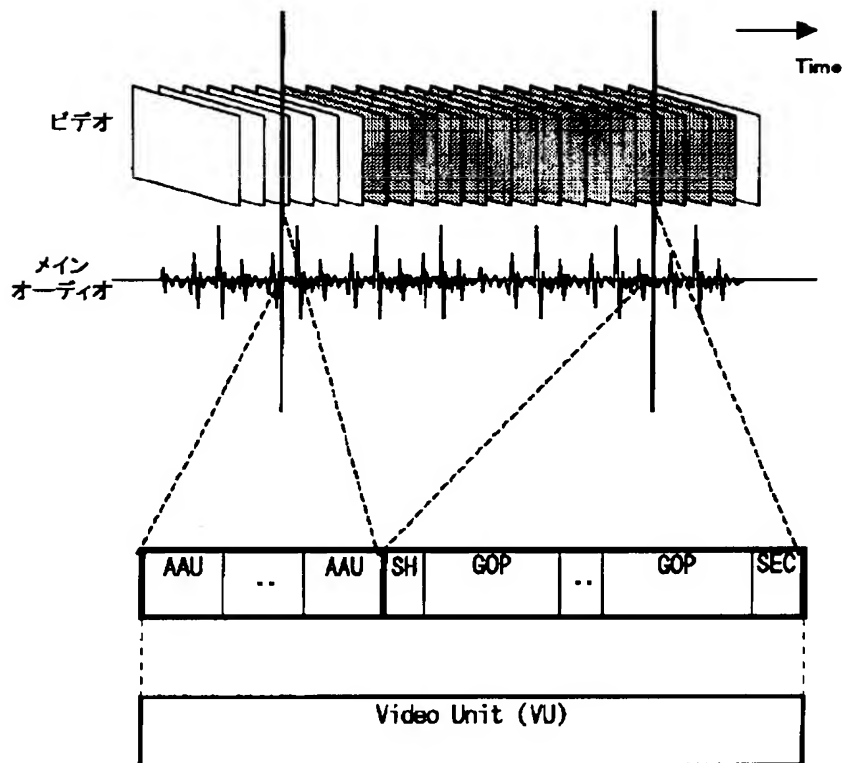
【図20】

```

Track fragment run atom {
  Atom size
  Type(='trun')
  version
  flags
  sample-count(=N)
  data-offset
  first-sample-flags
  for (i=0; i<N; i++){
    sample-duration
    sample-size
    sample-flags
    sample-composition-time-offset
  }
}

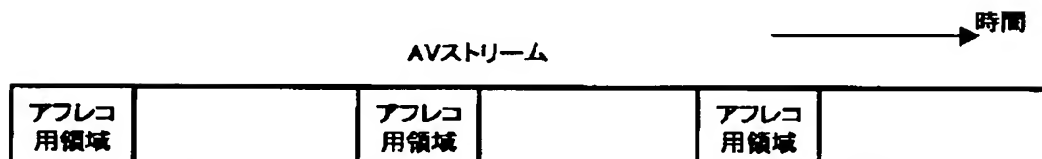
```

【図22】

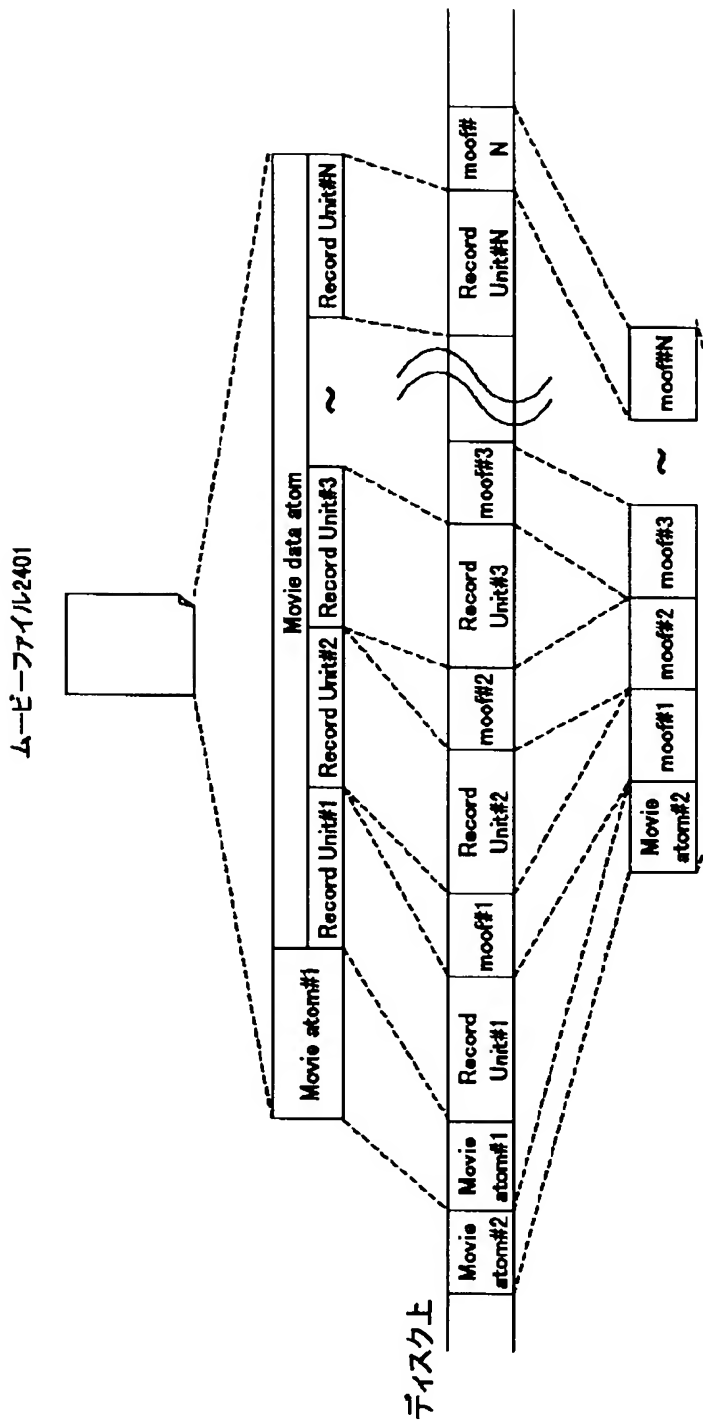


AAU: Audio Access Unit
 SH: Sequence Header
 SEC: Sequence End Code

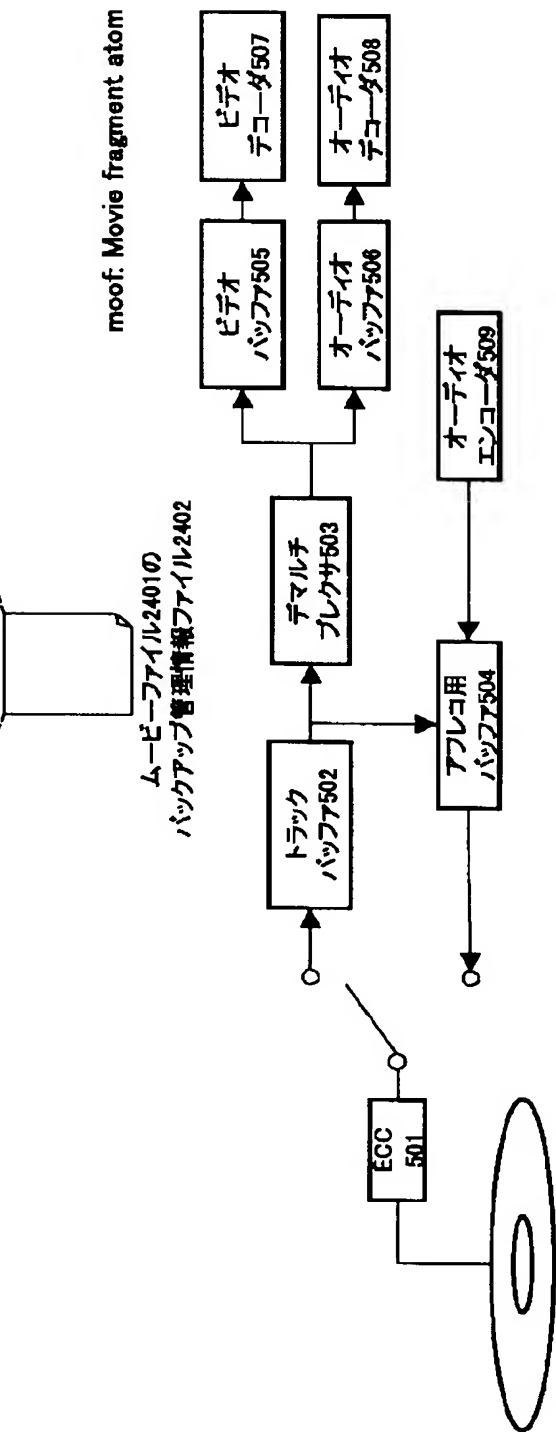
【図33】



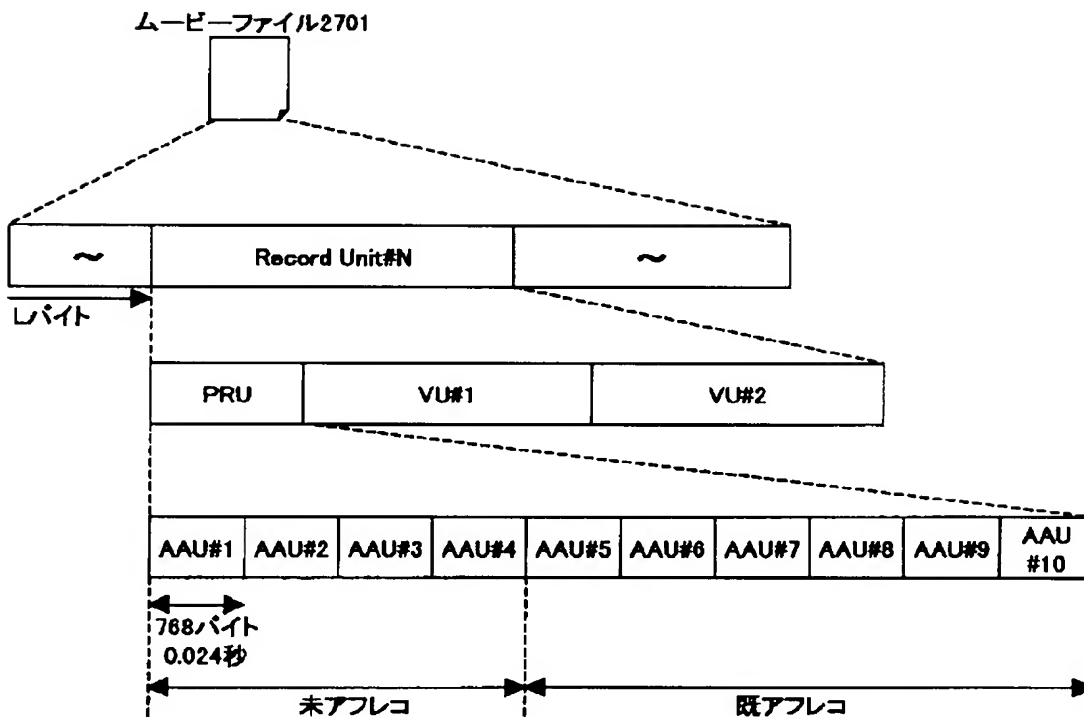
【図23】



【図28】



【図25】



【図26】

```

Movie atom {
  Movie extends atom {
    Track extends atom      /* ビデオトラック */
    Track extends atom      /* メインオーディオトラック */
    Track extends atom {    /* アフレコ領域トラック */
      default-sample-duration = 2160
      default-sample-size = 768
    }
    Track extends atom {    /* アフレコオーディオトラック */
      default-sample-duration = 2160
      default-sample-size = 768
    }
  }
}

```

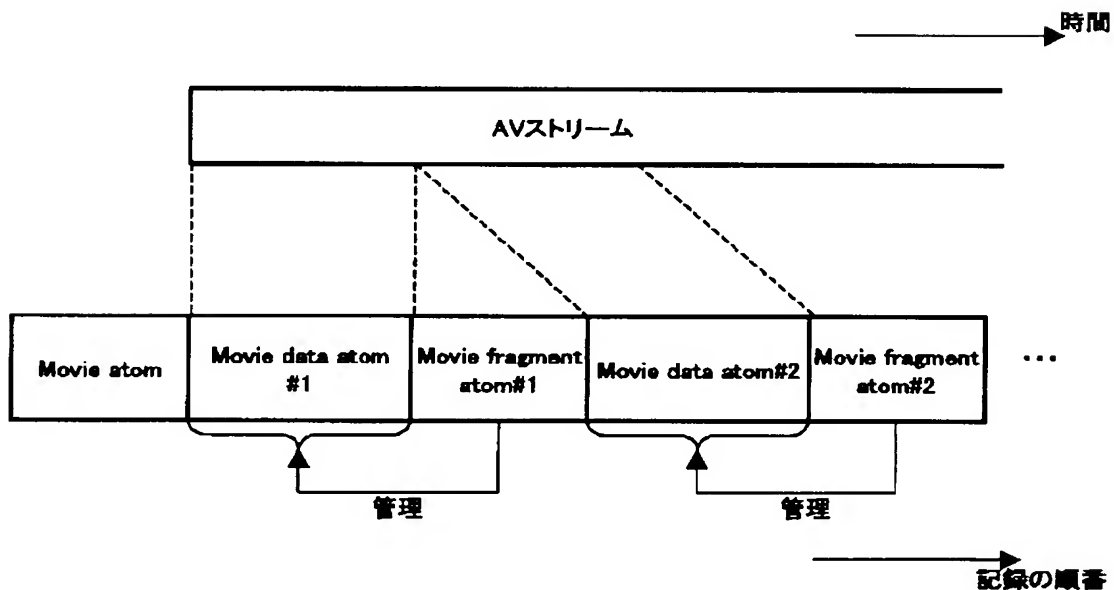
【図27】

```

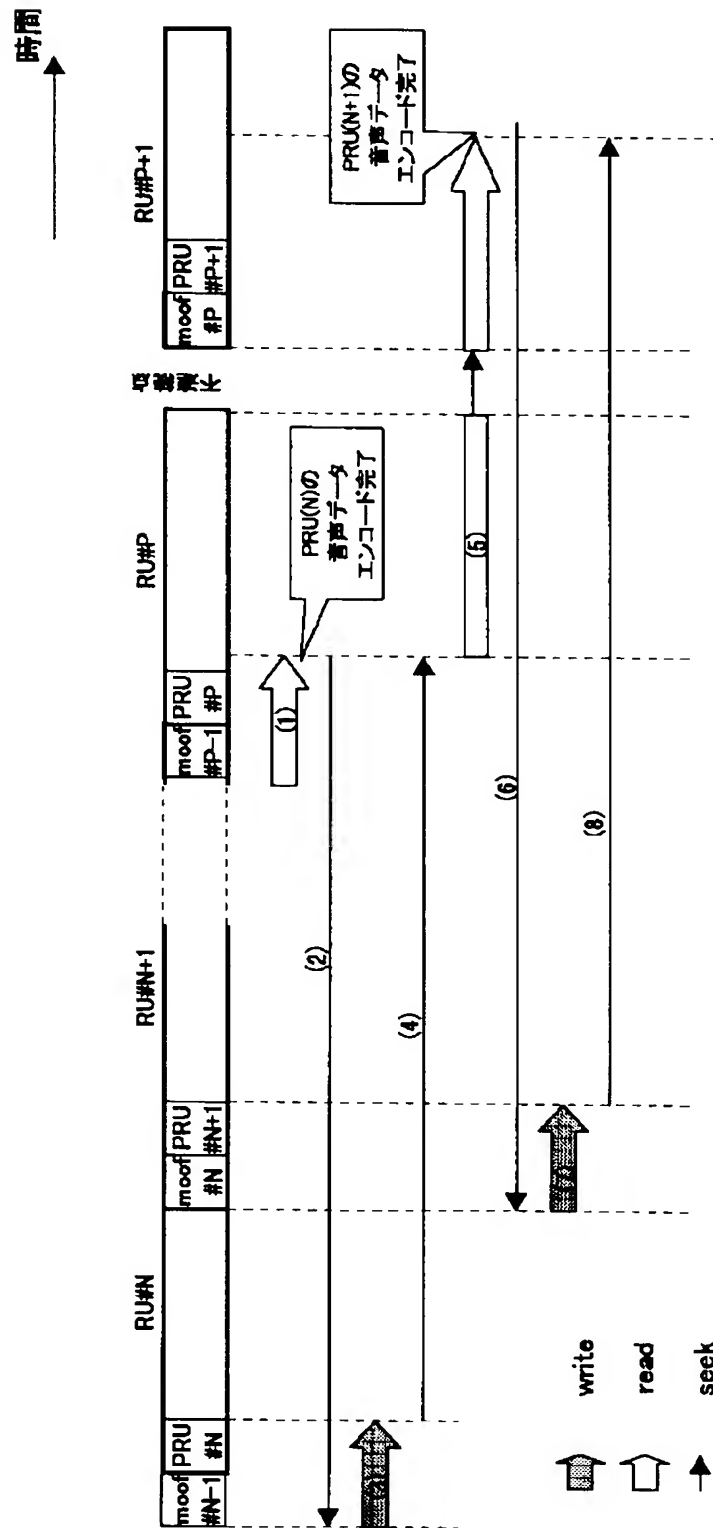
Movie fragment atom {
  Movie fragment header atom
  Track fragment atom /* ビデオトラック */
  Track fragment atom /* メインオーディオトラック */
  Track fragment atom { /* アフレコ領域トラック */
    Track fragment header atom {
      base-data-offset = L
    }
    Track fragment run atom #1 {
      sample-count = 10
      data-offset = 0
    }
  }
  Track fragment atom { /* アフレコオーディオトラック */
    Track fragment header atom {
      base-data-offset = L
    }
    Track fragment run atom #1 {
      sample-count = 6
      data-offset = 4 × 768
    }
  }
}

```

【図32】

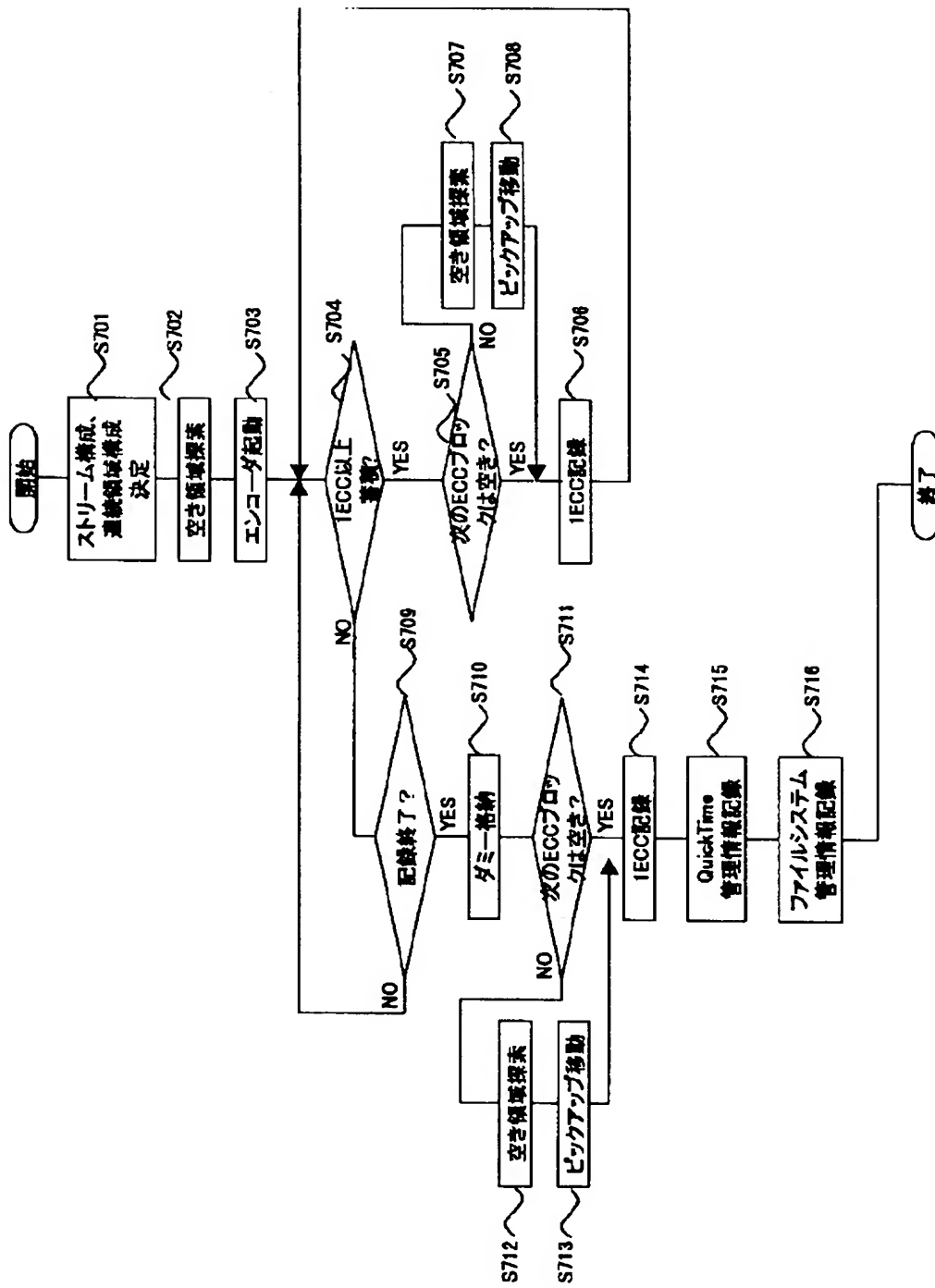


【図29】

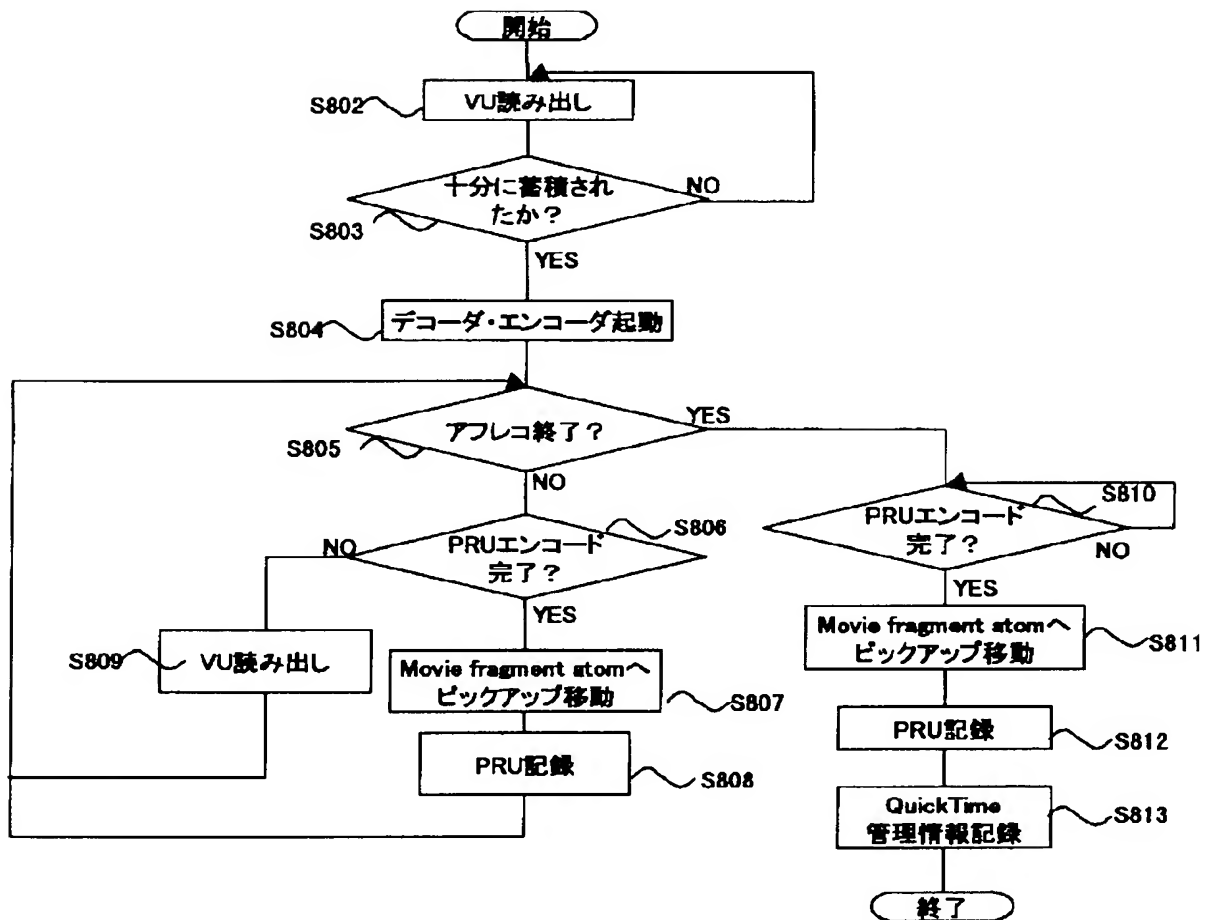


moof: Movie fragment atom

【図30】



【図31】



フロントページの続き

(72)発明者 山口 孝好
大阪府大阪市阿倍野区长池町22番22号 シ
ャープ株式会社内

Fターム(参考) 5C053 FA23 FA30 GA11 GB05 GB37
5D044 AB05 AB07 BC01 BC04 CC04
DE17 DE27 DE48 DE57 EF05
EF07
5D110 AA13 AA19 BB06 DA04 DA06
DA12 DA15 DB02 DD13